

Lecture Notes:

Computational Statistics

Part I. *From Maximum Likelihood to Generative Models*

Pr. Mohammed Nabil EL KORSO
mohammed.nabil.el-korso@centralesupelec.fr
Université Paris Saclay, L2S

CONTENTS

I	Abstract	3
II	A brief reminder of the MLE/MAP	3
II-A	Definitions	3
II-B	Link with the KLD	4
II-C	MLE through the SGD	7
III	The EM Algorithm	8
III-A	Definition & properties of the EM algorithm	8
III-B	Convergence analysis of the EM	10
III-C	Learning in latent variable models through the EM	12
III-D	Learning an optimal mixture of fixed models	13
IV	Beyond Standard EM	15
IV-A	Intractable E-step	16
IV-B	Intractable M-step	17
IV-C	Variational EM	18
V	Variational Autoencoders	18
V-A	Deep latent variable model	18
V-B	Linking the MLE to the ELBO	19
V-C	Linking the EM to the ELBO	20
V-D	Maximizing the ELBO through the SGD	20
VI	Appendix	23
VI-A	Other examples	23
VI-B	Other EM variants	35
	References	39

I. ABSTRACT

This lecture invites participants on a conceptual journey through probabilistic inference and latent variable modeling. We begin with the classical framework of Maximum Likelihood Estimation (MLE), the foundational pillar for parameter estimation in probabilistic models. Building on this, we revisit the Expectation-Maximization (EM) algorithm, a powerful iterative method for dealing with hidden data, highlighting its elegance and limitations in high-dimensional and complex systems. Our journey continues into the modern era of deep generative models with Variational Autoencoders (VAEs), where we explore how the principles of the EM naturally evolve within a variational inference framework. Special attention will be given to how VAEs approximate intractable posterior distributions and how this connects back to the EM philosophy. Throughout the lecture, applications in signal, image processing and machine learning will be considered.

II. A BRIEF REMINDER OF THE MLE/MAP

Let us denote the observation by $\mathbf{y} \in \mathbb{R}^{p \times 1}$ (the case of discrete or categorical observation can be deduced easily) and the true (unknown) probability function of the observation by $p^*(.)$. We assume that the data-generating process can be approximated by $p(.|\boldsymbol{\theta})$ in which $\boldsymbol{\theta} \in \Theta$ is an unknown parameter. The learning task consist in the finding $\boldsymbol{\theta}$ that leads to the smallest *distance* between $p(.|\boldsymbol{\theta})$ and $p^*(.)$.

In the case of observations that can be decomposed in covariates, $\mathbf{y}^{(1)} \in \mathbb{R}^{p_1 \times 1}$, and predictors $\mathbf{y}^{(2)} \in \mathbb{R}^{p_2 \times 1}$ with $p = p_1 + p_2$, the data-generating model can be conditional or joint model. Specifically,

- Discriminative models use the conditional model $p(\mathbf{y}^{(2)}|\mathbf{y}^{(1)}, \boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta})$ (e.g., classification, regression, ect.)
- Generative model use the joint model $p(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}|\boldsymbol{\theta})$ (e.g., Bayesian estimation, ect.)

In the following, we consider mainly the joint distribution, nevertheless, the adaptation to the conditional distribution is (in most cases) straightforward.

A. Definitions

A naturel way to find $\boldsymbol{\theta}$ is the Maximum Likelihood Estimator (MLE). The goal of the MLE is to find the optimal way to fit a distribution to the data, i.e., find

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \max_{\boldsymbol{\theta} \in \Theta} p(\mathbf{y}|\boldsymbol{\theta}) \quad (1)$$

Due to the monotonically increasing function of the log, it is common to maximize the log-likelihood. This allows some simplification in the numerical determination of the MLE as well leads to simplification of the study of the properties of the estimator. The latter reads

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \max_{\boldsymbol{\theta} \in \Theta} \log p(\mathbf{y}|\boldsymbol{\theta}) \quad (2)$$

under independence assumption:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \max_{\boldsymbol{\theta} \in \Theta} \log p(\mathbf{y}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta}) \quad (3)$$

By the nature of the objective function, the MLE is the estimator which makes the observed data most likely to occur. In other words, the MLE is the best *rationalization* of what we observed.

Before employing the MLE, it is necessary to check whether the data-generating process is sufficiently informative about the parameters of the model and that global identification is verified, namely, $\forall \boldsymbol{\theta}_1 \in \Theta$ and $\forall \boldsymbol{\theta}_2 \in \Theta$ with $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ implies that

$$\forall \mathbf{y} \in \mathcal{Y} \quad p(\mathbf{y}|\boldsymbol{\theta}_1) \neq p(\mathbf{y}|\boldsymbol{\theta}_2)$$

In the case of having a prior knowledge regarding the parameters, or in order to avoid singularities of the log-likelihood function (e.g., this a common scenario when learning a larger number of mixtures compared to the available data), we can make use of (or impose) some prior information. This is straightforwardly done by the MAP (*maximum a posteriori*) which reads as

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg \max_{\boldsymbol{\theta} \in \Omega} \log p(\boldsymbol{\theta}|\mathbf{y}) = \arg \max_{\boldsymbol{\theta} \in \Omega} (\log p(\mathbf{y}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})) \quad (4)$$

B. Link with the KLD

The MLE can be seen as the solution of the Kullback-Leibler divergence (KLD). Specifically, The KLD between two distributions writes

$$\text{KLD}(p(\mathbf{y})||q(\mathbf{y})) = \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y})} \left[\log \frac{p(\mathbf{y})}{q(\mathbf{y})} \right] = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} \quad (5)$$

with $\text{KLD}(p(\mathbf{y})||q(\mathbf{y})) \neq \text{KLD}(q(\mathbf{y})||p(\mathbf{y}))$, $\text{KLD}(p(\mathbf{y})||q(\mathbf{y})) \geq 0$ and $\text{KLD}(p(\mathbf{y})||q(\mathbf{y})) = 0 \Leftrightarrow p(\mathbf{y}) = q(\mathbf{y})$.

Given a training set $\{\mathbf{y}_i\}_i$, with $\mathbf{y}_i \sim p^*(\mathbf{y})$, it can be seen that the MLE minimizes the KLD:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\text{KLD}} &= \arg \min_{\boldsymbol{\theta} \in \Theta} \text{KLD}(p^*(\mathbf{y})||p(\mathbf{y}|\boldsymbol{\theta})) = \arg \min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\mathbf{y} \sim p^*(\mathbf{y})} \left[\log \frac{p^*(\mathbf{y})}{p(\mathbf{y}|\boldsymbol{\theta})} \right] \\ &= \arg \max_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\mathbf{y} \sim p^*(\mathbf{y})} [\log p(\mathbf{y}|\boldsymbol{\theta})] \approx \arg \max_{\boldsymbol{\theta} \in \Theta} \frac{1}{n} \sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}_{\text{MLE}}. \end{aligned}$$

Example 2.1: [MLE of mean and covariance matrix in multivariate Gaussian observations.]

Let $\{\mathbf{y}_i\}_{i=1}^n$ be a set of independent and identically distributed (i.i.d.) observations with

$$\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (6)$$

where $\mathbf{y}_i \in \mathbb{R}^p$, $\boldsymbol{\mu} \in \mathbb{R}^p$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ is a symmetric positive-definite matrix. The probability density function of $\mathbf{y}_i$ reads

$$p(\mathbf{y}_i | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}) \right).$$

Since the observations are independent, the log likelihood, $\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, reduces to

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^n \log p(\mathbf{y}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto -\frac{n}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\mu}).$$

To find the MLE of $\boldsymbol{\mu}$, take the derivative of $\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\mu}$

$$\frac{\partial \ell(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1} \sum_{i=1}^n (\mathbf{y}_i - \boldsymbol{\mu}).$$

Setting this derivative to zero:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i \quad (7)$$

Plugging (7) in (6) and taking the derivative of $\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\Sigma}$ reads

$$\frac{\partial \ell(\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} = -\frac{n}{2} \boldsymbol{\Sigma}^{-1} + \frac{1}{2} \sum_{i=1}^n \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}) (\mathbf{y}_i - \hat{\boldsymbol{\mu}})^\top \boldsymbol{\Sigma}^{-1}$$

Setting this to zero

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \hat{\boldsymbol{\mu}}) (\mathbf{y}_i - \hat{\boldsymbol{\mu}})^\top. \quad (8)$$

□

Unfortunately, the MLE can be always found in closed form expression as in Example 2.1 as shown in the next toy example.

Example 2.2: (Learning an optimal mixture of fixed models)

We consider here the learning problem of an optimal combination of arbitrary known models. Let us suppose that we observe $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\top$ that might be generated using k models $(p_1, \dots, p_k)$. In addition, we suppose that the observation are i.i.d. from a combination of these k models, such that

$$p(\mathbf{y}_i | \boldsymbol{\theta}) = \sum_{j=1}^k \theta_j p_j(\mathbf{y}_i) \quad (9)$$

in which $\sum_{j=1}^k \theta_j = 1$ and $\theta_j \in [0, 1]$, $\forall j$. Our aim is to learn (estimate) the most likely combination $\boldsymbol{\theta} = [\theta_1, \dots, \theta_k]^T$ of models to characterize the observed data $\mathbf{y}$.

In this case, the maximum likelihood reads

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^n \log \sum_{j=1}^k \theta_j p_j(\mathbf{y}_i) \quad (10)$$

It can be seen from (10) that the MLE of $\boldsymbol{\theta}$ does not admit a closed form (this is mainly due to the summation inside the Log). Solving (10) can be computational heavy even in such simple scenario for large dimension of $\boldsymbol{\theta}$.

It got worst in a more realistic scenario when we are in the presence of unknown models, e.g., in the Gaussian mixture scenario, p_j denotes a Gaussian pdf with unknown mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, i.e., $p_j(\mathbf{y}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, the maximum likelihood becomes

$$\hat{\boldsymbol{\theta}}, \{\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j\}_j = \underset{\boldsymbol{\theta}, \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_j}{\operatorname{argmax}} \sum_{i=1}^n \log \sum_{j=1}^k \theta_j \mathcal{N}(\mathbf{y}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \quad (11)$$

A closed form expression of (11) is only possible in specific and restrictive scenarios, as when $k = 1$ (in which the summation inside the Log disappear), e.g. Example (2.1). $\square$

From Example 2.2 it is clear that knowing which model is activated will simplify considerably the problem. If a genius give us this indication for each observation $\mathbf{y}_i$, then, the MLE reads

$$\begin{aligned} \hat{\theta}_j &= \frac{n_j}{n} \\ \hat{\boldsymbol{\mu}}_j &= \frac{1}{n_j} \sum_{i=1}^n \mathbf{y}_i \mathbb{1}_{j \rightarrow i} \\ \hat{\boldsymbol{\Sigma}}_j &= \frac{1}{n_j} \sum_{i=1}^n (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_j)^\top \mathbb{1}_{j \rightarrow i} \end{aligned}$$

in which $\mathbb{1}_{j \rightarrow i} = 1$ if the j -th model is the generating process for the i -th observation, otherwise $\mathbb{1}_{j \rightarrow i} = 0$ and $n_j = \sum_i \mathbb{1}_{j \rightarrow i}$.

Unfortunately, it is common to not have a genius in our pocket. The EM algorithm tries to guess what such a genius might have told us. The EM algorithm is particularly useful in situations where additional, but unavailable, information (known as latent or hidden variables) is part of the model, and where these variables would significantly simplify the learning task if they were observed.

C. MLE through the SGD

We recall that under independence assumption, the MLE writes

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \max_{\boldsymbol{\theta} \in \Theta} \log p(\mathbf{y}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta}) \quad (12)$$

In the case of non tractable MLE (impossibility of having closed form), one can use chain rule and automatic differentiation tools in order to compute gradients of the objection function, namely, of $l(\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^n \log p(\mathbf{y}_i|\boldsymbol{\theta})$. Computing the gradient for each sample, gives us the so-called *batch* gradient descent. This can be heavy for large n . Instead the so-called *stochastic* gradient descent. The latter reads

$$\boldsymbol{\theta}^{(m+1)} \leftarrow \boldsymbol{\theta}^{(m)} + \eta_m \nabla_{\boldsymbol{\theta}} l(\tilde{\boldsymbol{\theta}}|\boldsymbol{\varepsilon}) \quad (13)$$

in which η_m denotes the learning rate and $\tilde{l}(\boldsymbol{\theta}|\boldsymbol{\varepsilon})$ is unbiased estimate of $l(\boldsymbol{\theta}|\mathbf{y})$, namely,

$$E_{\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})} \left\{ \tilde{l}(\boldsymbol{\theta}|\boldsymbol{\varepsilon}) \right\} = l(\boldsymbol{\theta}|\mathbf{y}). \quad (14)$$

The random variable $\boldsymbol{\varepsilon}$ can be tailored as a random variable that selects (randomly and uniformly) minibatches $\mathcal{M}_{\text{mini}}$ of size n_{mini} . With this minibatches, we have unbiased estimator of the gradient, i.e.,

$$\frac{1}{n_{\text{mini}}} \sum_{i, \mathbf{y}_i \in \mathcal{M}_{\text{mini}}} \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\mathbf{y}_i)$$

is unbiased estimator of

$$\frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\mathbf{y}_i).$$

Pseudo code for the SGD

- *Initialization:* Choose initial estimate $\boldsymbol{\theta}^{(0)}$ and collection of learning rates $\{\eta_m\}$.
- Repeat until convergence
 - Randomly chose a minibatch $\mathcal{M}_{\text{mini}}^{(m)}$ of size n_{mini}
 - Update

$$\boldsymbol{\theta}^{(m+1)} \leftarrow \boldsymbol{\theta}^{(m)} + \frac{\eta_m}{n_{\text{mini}}} \sum_{i, \mathbf{y}_i \in \mathcal{M}_{\text{mini}}^{(m)}} \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\mathbf{y}_i)$$

III. THE EM ALGORITHM

A. Definition & properties of the EM algorithm

1) *Definition:* The EM algorithm needs addition dataset named complete data $\mathbf{x}$. Their parametric density is denoted $p(\mathbf{x}|\boldsymbol{\theta})$. A guide line to chose the complete data is given after. It is important to be aware that we do not observe the complete data $\mathbf{x}$, we observe only $\mathbf{y}$ that should depend on $\mathbf{x}$. The EM algorithm is mainly constituted by the two steps: *Expectation* and *Maximization* steps as follows

- *Initialization:* For $m = 0$, make an initial estimate of $\boldsymbol{\theta}^{(0)}$ (using a prior knowledge or using a sub optimal existing algorithm)
- *E-step:* Formulate the pdf $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(m)})$ based on the previously estimated $\boldsymbol{\theta}^{(m)}$. Then, compute

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= \int \log p(\mathbf{x}|\boldsymbol{\theta}) p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(m)}) d\mathbf{x} \\ &= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{x}|\boldsymbol{\theta}) \} \end{aligned} \quad (15)$$

- *M-step:* The $m + 1$ -th estimate of $\boldsymbol{\theta}$ reads as

$$\boldsymbol{\theta}^{(m+1)} = \arg \max_{\boldsymbol{\theta} \in \Theta} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \quad (16)$$

- *Incrementing:* Increment m and repeat the above procedure (E-step and M-step) until a stopping criteria is met.

It is worth mentioning that the EM estimate is only guaranteed to never get worse (we discuss this later on). Usually, it will find a peak in the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$, but if the likelihood function has multiple peaks, EM will not necessarily find the global maximum of the likelihood. In practice, it is common to start the EM from multiple random initial guesses, and choose the one with the largest likelihood as the final guess for $\boldsymbol{\theta}$.

2) *Properties of the complete data:* The first element that one should keep in mind is that specifying the complete data X must lead to a relatively easy maximization of $p(\mathbf{x}|\boldsymbol{\theta})$ for a given $\mathbf{x}$ with respect to $\boldsymbol{\theta}$. In addition, the complete data X must satisfy the so-called following Markov relationship

$$\boldsymbol{\theta} \rightarrow X \rightarrow Y \quad (17)$$

meaning that

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{x}) \quad (18)$$

As an example, when the complete data is related to the data via a deterministic function, i.e., $Y = F(X)$, with $F(\cdot)$ a deterministic function, then (17) holds naturally.

3) *EM formation in the case of i.i.d. samples*: In many applications it is common to consider the observed data to be i.i.d. (independently and identically distributed samples). In this case, it is of interest to consider the complete data as i.i.d. such that the i -th observation $\mathbf{y}_i$ is only a function of $\mathbf{x}_i$. This leads to the following proposition

Lemma 3.1: Let us assume that

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^n p(\mathbf{x}_i|\boldsymbol{\theta})$$

with $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$ and the Markov relationship reads

$$\boldsymbol{\theta} \rightarrow X_i \rightarrow Y_i, \quad \forall i = 1, \dots, n$$

that is

$$p(\mathbf{y}_i|\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_{i-1}, \mathbf{y}_{i+1}, \dots, \mathbf{y}_n, \boldsymbol{\theta}) = p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}) \quad (19)$$

then

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = \sum_{i=1}^n Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \quad (20)$$

in which

$$Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = E_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{x}_i|\boldsymbol{\theta}) \}, \quad \forall i = 1, \dots, n$$

□

Proof: First, let us prove the mutual independence. We denote $\mathbf{y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$,

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = p(\mathbf{y}_1|\mathbf{y}_2, \dots, \mathbf{y}_n, \mathbf{x}, \boldsymbol{\theta}) \dots p(\mathbf{y}_n|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})$$

by the chaine rule

$$= p(\mathbf{y}_1|\mathbf{x}_1, \boldsymbol{\theta}) \dots p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})$$

by (19)

$$\begin{aligned} &= p(\mathbf{y}_1|\mathbf{x}_1, \boldsymbol{\theta}) \dots p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\theta}) \prod_{i=1}^n p(\mathbf{x}_i|\boldsymbol{\theta}) \\ &= \prod_{i=1}^n p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}) p(\mathbf{x}_i|\boldsymbol{\theta}) \\ &= \prod_{i=1}^n p(\mathbf{x}_i, \mathbf{y}_i|\boldsymbol{\theta}) \end{aligned} \quad (21)$$

Second, we notice that

$$\begin{aligned}
p(\mathbf{x}_i|\mathbf{y},\boldsymbol{\theta}) &= \frac{p(\mathbf{x}_i,\mathbf{y}|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta})} \\
&= \frac{\int_{\mathcal{X}^{n-1}} p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) d\mathbf{x}_1 \dots d\mathbf{x}_{i-1} d\mathbf{x}_{i+1} \dots d\mathbf{x}_n}{\int_{\mathcal{X}^n} p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta}) d\mathbf{x}} \\
&= \frac{\int_{\mathcal{X}^{n-1}} \prod_{j=1}^n p(\mathbf{x}_j,\mathbf{y}_j|\boldsymbol{\theta}) d\mathbf{x}_1 \dots d\mathbf{x}_{i-1} d\mathbf{x}_{i+1} \dots d\mathbf{x}_n}{\int_{\mathcal{X}^n} \prod_{j=1}^n p(\mathbf{x}_j,\mathbf{y}_j|\boldsymbol{\theta}) d\mathbf{x}_1 \dots d\mathbf{x}_n} \\
&\quad \text{by (21)} \\
&= \frac{p(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\theta}) \prod_{j=1,j \neq i}^n \int_{\mathcal{X}} p(\mathbf{x}_j,\mathbf{y}_j|\boldsymbol{\theta}) d\mathbf{x}_j}{\prod_{j=1}^n \int_{\mathcal{X}} p(\mathbf{x}_j,\mathbf{y}_j|\boldsymbol{\theta}) d\mathbf{x}_j} \\
&= \frac{p(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\theta}) \prod_{j=1,j \neq i}^n p(\mathbf{y}_j|\boldsymbol{\theta})}{\prod_{j=1}^n p(\mathbf{y}_j|\boldsymbol{\theta})} \\
&= \frac{p(\mathbf{x}_i,\mathbf{y}_i|\boldsymbol{\theta})}{p(\mathbf{y}_i|\boldsymbol{\theta})} \\
&= p(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\theta})
\end{aligned} \tag{22}$$

Consequently, using the independence assumption of X and (22) in the last equality, one obtains

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{x}|\boldsymbol{\theta}) \} \\
&= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}^{(m)})} \left\{ \log \prod_{i=1}^n p(\mathbf{x}_i|\boldsymbol{\theta}) \right\} \\
&= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}^{(m)})} \left\{ \sum_{i=1}^n \log p(\mathbf{x}_i|\boldsymbol{\theta}) \right\} \\
&= \sum_{i=1}^n E_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{y},\boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{x}_i|\boldsymbol{\theta}) \} \\
&= \sum_{i=1}^n E_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{y},\boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{x}_i|\boldsymbol{\theta}) \}
\end{aligned}$$

B. Convergence analysis of the EM

The following monotonicity property of the EM algorithm, states that improving the Q function will at least not make the log-likelihood $l(\boldsymbol{\theta})$ worse:

Theorem 3.2: [Convergence the EM]

Let the random variables X and Y have parametric densities w.r.t. the parameter $\boldsymbol{\theta} \in \Theta$. Suppose the support of X does not depend on $\boldsymbol{\theta}$, and the Markov relationship $\boldsymbol{\theta} \rightarrow X \rightarrow Y$ holds, that is

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{x}) \tag{23}$$

holds for all $\theta \in \Theta$, $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. Then for $\theta \in \Theta$ and any $\mathbf{y} \in \mathcal{Y}$ with $\mathcal{X}(\mathbf{y}) \neq \emptyset$, $l(\theta) \geq l(\theta^{(m)})$ if $Q(\theta|\theta^{(m)}) \geq Q(\theta^{(m)}|\theta^{(m)})$.

Proof: Basically, the proof uses the definition of the likelihood combined with the Markov relationship and the law of total probability. Specifically,

$$\begin{aligned}
l(\theta) &= \log p(\mathbf{y}|\theta) = \log \int_{\mathcal{X}(\mathbf{y})} p(\mathbf{x}, \mathbf{y}|\theta) d\mathbf{x} \\
&= \log \int_{\mathcal{X}(\mathbf{y})} \frac{p(\mathbf{x}, \mathbf{y}|\theta)}{p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} p(\mathbf{x}|\mathbf{y}, \theta^{(m)}) d\mathbf{x} \\
&= \log E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \left\{ \frac{p(\mathbf{x}, \mathbf{y}|\theta)}{p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \right\} \\
&\geq E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \left\{ \log \frac{p(\mathbf{x}, \mathbf{y}|\theta)}{p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \right\} \\
&\quad \text{using Jensen's inequality}^1 \\
&= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \left\{ \log \frac{p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x})}{p(\mathbf{x}|\theta^{(m)})p(\mathbf{y}|\mathbf{x})/p(\mathbf{y}|\theta^{(m)})} \right\} \\
&\quad \text{using Bayes' rule and the assumed Markov relationship}^2 \\
&= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \left\{ \log \frac{p(\mathbf{x}|\theta)p(\mathbf{y}|\theta^{(m)})}{p(\mathbf{x}|\theta^{(m)})} \right\} \\
&= E_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \theta^{(m)})} \{ \log p(\mathbf{x}|\theta) \} - E_{\mathbf{x}|\mathbf{y}, \theta^{(m)}} \{ \log p(\mathbf{x}|\theta^{(m)}) \} + \log p(\mathbf{y}|\theta^{(m)}) \\
&= Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)}) + l(\theta^{(m)})
\end{aligned}$$

Meaning that

$$l(\theta) \geq Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)}) + l(\theta^{(m)}) \quad (24)$$

Consequently, choosing θ that maximizes $Q(\theta|\theta^{(m)})$ leads to $Q(\theta|\theta^{(m)}) \geq Q(\theta^{(m)}|\theta^{(m)})$, then,

$$l(\theta) \geq Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)}) + l(\theta^{(m)}) \geq l(\theta^{(m)}) \quad (25)$$

which concludes the proof. $\square$

Remark 3.3:

¹Jensen's inequality: for f convex $f(\sum_i \lambda_i \mathbf{x}_i) \leq \sum_i \lambda_i f(\mathbf{x}_i)$

²Using the fact that, for the numerator $p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\mathbf{y}|\mathbf{x})$ and for the denominator $p(\mathbf{x}|\mathbf{y}, \theta^{(m)}) = \frac{p(\mathbf{y}|\mathbf{x}, \theta^{(m)})p(\mathbf{x}|\theta^{(m)})}{p(\mathbf{y}|\theta^{(m)})} = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\theta^{(m)})}{p(\mathbf{y}|\theta^{(m)})}$

- From the M-step

$$\boldsymbol{\theta}^{(m+1)} = \arg \max_{\boldsymbol{\theta} \in \Theta} Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})$$

Consequently, $Q(\boldsymbol{\theta}^{(m+1)} | \boldsymbol{\theta}^{(m)}) \geq Q(\boldsymbol{\theta}^{(m)} | \boldsymbol{\theta}^{(m)})$ and by applying Theorem 3.2, one obtains $l(\boldsymbol{\theta}^{(m+1)}) \geq l(\boldsymbol{\theta}^{(m)})$

- The monotonicity of the EM algorithm guarantees that as EM iterates, its guesses won't get worse in terms of their likelihood, but the monotonicity alone cannot guarantee the convergence of the sequence $\{\boldsymbol{\theta}^{(m)}\}$
- Specifically, if $l(\boldsymbol{\theta})$ is bounded from above on Θ , then the monotonicity implies the convergence of the sequence $\{l(\boldsymbol{\theta}^{(m)})\}$, but not of the sequence $\{\boldsymbol{\theta}^{(m)}\}$.

Remark 3.4: Including a Prior in the EM algorithm

As explained above, in order to enhance the estimation process when one has a prior knowledge, or in order to avoid singularities of the log-likelihood function (e.g., this a common scenario when learning a larger number of mixtures compared to the available data), we can make use of (or impose) some prior information. This is straightforwardly done by the MAP (*maximum a posteriori*) which reads as

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg \max_{\boldsymbol{\theta} \in \Omega} \log p(\boldsymbol{\theta} | \mathbf{y}) = \arg \max_{\boldsymbol{\theta} \in \Omega} (\log p(\mathbf{y} | \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})) \quad (26)$$

Consequently, the EM algorithm can be straightforwardly extended as

- *E-step*: Given $\boldsymbol{\theta}^{(m)}$ estimate

$$Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)}) = E_{\mathbf{x} \sim p(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^{(m)})} \{\log p(\mathbf{x} | \boldsymbol{\theta})\} \quad (27)$$

- *M-step*: The maximization step reads

$$\boldsymbol{\theta}^{(m+1)} = \arg \min_{\boldsymbol{\theta} \in \Theta} \left(Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)}) + \log p(\boldsymbol{\theta}) \right) \quad (28)$$

The monotonicity of the MAP EM is proved following the same lines of Theorem 3.2.

C. Learning in latent variable models through the EM

Following the same methodology, the EM algorithm can be employed to estimate the parameters of directed models that involve latent variables. These latent variables, denoted by $\mathbf{z}$, are incorporated into the model's structure but are not directly observed in the available data. When performing unconditional modeling of an observed variable $\mathbf{y}$, the corresponding directed graphical model describes a joint

probability distribution $p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta})$ involving both the observed data and the hidden variables. To obtain the distribution over the observed variable alone, one marginalizes over the latent variables, resulting

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \quad (29)$$

The latter is referred to marginal likelihood (in the context of MLE estimation) or the model evidence (in the context of generative models), when taken as a function of $\boldsymbol{\theta}$. In this case, the complete data are given as $\mathbf{x} = \{\mathbf{y}, \mathbf{z}\}$ and the EM algorithm can be used to learn $\boldsymbol{\theta}$. In this case, the E-step reads

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= \int \log p(\mathbf{x}|\boldsymbol{\theta}) p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(m)}) d\mathbf{x} \\ &= \int \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) p(\mathbf{y}, \mathbf{z}|\mathbf{y}, \boldsymbol{\theta}^{(m)}) d\mathbf{x} \\ &= \int \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta}^{(m)}) d\mathbf{z} \\ &= E_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) \} \\ &= E_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta}^{(m)})} \{ \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) \} \end{aligned} \quad (30)$$

In the context of deep latent variable model (DLVM) distributions involved in $p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z}|\boldsymbol{\theta})p(\mathbf{y}|\mathbf{z}, \boldsymbol{\theta})$ are parameterized by neural networks. It is of interest, since even if we assume relatively simple distribution (e.g., parameterized Gaussian distribution) the marginal distribution $p(\mathbf{y}|\boldsymbol{\theta})$ can be very complex. This expressivity makes deep latent-variable models attractive for approximating complicated underlying distributions $p^*(\mathbf{y})$.

D. Learning an optimal mixture of fixed models

As a first and simple example, we consider here the learning problem of an optimal combination of arbitrary known models. This will be extended later in the case of unknown models and applied in the context of robust array signal processing (robust direction of arrivals estimation) [1].

Let us suppose that we observe $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^T$ that might be generated using k models $(p_1, \dots, p_k)$. As an example, in the Gaussian mixture scenario, p_i denotes a Gaussian pdf with known mean and covariance matrix. In addition, we suppose that the observation are i.i.d. from a combination of these k models, such that

$$p(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{j=1}^k \theta_j p_j(\mathbf{y}_j) \quad (31)$$

in which $\sum_{j=1}^k \theta_j = 1$ and $\theta_j \in [0, 1]$, $\forall j$. Our aim is to learn (estimate) the most likely combination $\boldsymbol{\theta} = [\theta_1, \dots, \theta_k]^T$ of models to characterize the observed data $\mathbf{y}$.

It is clear that knowing which model is activated will simplify considerably the problem. Thus, it is natural to choose the EM complete data as the observed data, plus, a set of hidden data $\mathbf{z} = [z_1, \dots, z_n]^T$ representing which of the k models are used to generate the observation. That is $z_i \in \{1, \dots, k\}$ with $i \in \{1, \dots, n\}$. This leads to

$$p(\mathbf{y}_i, z_i = j | \boldsymbol{\theta}) = \theta_j p_j(\mathbf{y}_i) \quad (32)$$

Consequently, from (32), one obtains easily

$$P(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) = \frac{p(z_i = j, \mathbf{y}_i | \boldsymbol{\theta}^{(m)})}{p(\mathbf{y}_i | \boldsymbol{\theta}^{(m)})} = \frac{\theta_j^{(m)} p_j(\mathbf{y}_i)}{\sum_{l=1}^k \theta_l^{(m)} p_l(\mathbf{y}_i)} \quad (33)$$

Then, the M-step reduces to

$$\begin{aligned} \boldsymbol{\theta}^{(m)} &= \arg \max_{\boldsymbol{\theta} \in \Theta} E_{\mathbf{z} | \mathbf{y}, \boldsymbol{\theta}^{(m)}} \{ \log p(\mathbf{y}, \mathbf{z} | \boldsymbol{\theta}) \}, \text{ using the EM formulation for missing data} \\ &= \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \sum_{j=1}^k p(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) \{ \log p(\mathbf{y}_i, z_i = j | \boldsymbol{\theta}) \}, \text{ using the EM formulation for i.i.d. data} \\ &= \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \sum_{j=1}^k p(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) \{ \log \theta_{z_i=j} + \log p_{z_i=j}(\mathbf{y}_i) \} \quad \text{using (31)} \\ &= \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \sum_{j=1}^k p(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) \log \theta_j \\ &= \arg \max_{\boldsymbol{\theta} \in \Theta} \sum_{j=1}^k \alpha_j^{(m)} \log \theta_j \end{aligned}$$

in which, from (33), the above reduces to

$$\alpha_j^{(m)} = \sum_{i=1}^n p(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) = \sum_{i=1}^n \frac{\theta_j^{(m)} p_j(\mathbf{y}_i)}{\sum_{l=1}^k \theta_l^{(m)} p_l(\mathbf{y}_i)}.$$

Solving the constrained problem above can be easily handled by the method of Lagrange multipliers. For now, we ignore the constraint $\theta_j \in [0, 1]$, $\forall j$, and we consider only the constraint $\sum_j \theta_j = 1$. This leads to

$$\frac{\partial}{\partial \theta_j} \left(\sum_{l=1}^k \alpha_l^{(m)} \log \theta_l - \lambda \left(\sum_{l=1}^k \theta_l - 1 \right) \right) = \frac{\alpha_j^{(m)}}{\theta_j} - \lambda$$

equating the above to zero, leads to $\theta_j^* = \frac{\alpha_j^{(m)}}{\lambda}$. Now, we choose λ that naturally satisfies $\sum_j \theta_j = 1$, i.e., this leads to $\theta_j^* = \frac{\alpha_j^{(m)}}{\sum_{l=1}^k \alpha_l^{(m)}}$. Finally, note that the problem in (33) is actually a convex optimization problem, and here we skip the details of verifying the optimality of the solution.

Finally, the EM reduces to

EM for learning an optimal mixture of fixed models

- *Initialization:* Choose initial estimate $\boldsymbol{\theta}^{(0)}$
- *E-step:* Compute

$$\alpha_j^{(m)} = p(z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) = \sum_{i=1}^n \frac{\theta_j^{(m)} p_j(\mathbf{y}_i)}{\sum_{l=1}^k \theta_l^{(m)} p_l(\mathbf{y}_i)}, \quad j = 1, \dots, k$$

- *M-step:* Update

$$\theta_j^{(m+1)} = \frac{\alpha_j^{(m)}}{\sum_{l=1}^k \alpha_l^{(m)}}, \quad j = 1, \dots, k$$

- *Convergence check:* Compute $l(\boldsymbol{\theta}^{(m+1)})$, if $|l(\boldsymbol{\theta}^{(m+1)}) - l(\boldsymbol{\theta}^{(m)})| > \delta$, for a chosen δ , return to E-step, otherwise end the algorithm
-

IV. BEYOND STANDARD EM

The EM algorithm is a powerful and widely-used technique for maximum likelihood estimation when dealing with incomplete or latent data. Its iterative process alternates between estimating missing information and optimizing model parameters, making it conceptually simple and broadly applicable [2]–[4].

However, despite its strengths, the standard EM algorithm faces practical limitations. In particular, *i)* it can converge slowly, especially near flat regions of the likelihood surface, *ii)* it is often sensitive to initial parameter values, risking convergence to poor local optima, and, *iii)* in many problems of interest, especially those involving complex latent structures, or large datasets, either the expectation (E-step) or maximization (M-step) becomes computationally intractable. These limitations have motivated the development of several variants and extensions of the EM algorithm, designed to improve convergence behavior, reduce computational cost, or handle situations where exact updates are impossible. In this section, we will see how the core EM framework can be adapted and extended to tackle these challenges while preserving its fundamental structure.

A. Intractable E-step

If the E-step is intractable, a natural choice is to use a stochastic variant of the EM algorithm. Namely, instead of computing the full expectation, it approximates it using random samples, often via Monte Carlo methods. This allows the algorithm to proceed even when the exact E-step is infeasible (e.g., due to a complex probability density function or in non linear settings), enabling efficient parameter estimation while maintaining convergence properties under certain conditions. Several stochastic versions of the EM algorithm have been developed to address such kind of situation. To cite a few,

- Stochastic EM (SEM) replaces the full E-step with a sample-based approximation, often using a single or a few random draws from the conditional distribution of the latent variables. This allows the algorithm to handle large or complex datasets when computing the exact expectation is impractical.
- Monte Carlo EM (MCEM) improves this by using Monte Carlo sampling to approximate the E-step with multiple samples, providing a more accurate estimate of the expectation. It is especially useful when analytical solutions are unavailable but sampling is feasible.
- Stochastic approximation EM (SAEM) goes further by combining stochastic updates with a smoothing mechanism, updating the expectation incrementally over iterations. This leads to better stability and convergence, even when using fewer samples at each iteration [5].

In Table. I presents the main stochastic EM variants.

	E-Step Description	Analytical Equation
EM	Computes the exact conditional expectation of the complete-data log-likelihood.	$Q(\theta \theta^{(m)}) = \mathbb{E}_{\mathbf{x} \sim f(\mathbf{x} \mathbf{y}, \theta^{(m)})} [\log f(\mathbf{x} \theta)]$
SEM	Approximates the E-step using a single sample from the posterior.	$Q(\theta \theta^{(m)}) \approx \log f(\mathbf{x}^{(m)} \theta)$ where $\mathbf{x}^{(m)} \sim f(\mathbf{x} \mathbf{y}, \theta^{(m)})$
MCEM	Uses multiple samples to estimate the expectation in the E-step.	$Q(\theta \theta^{(m)}) \approx \frac{1}{K} \sum_{c=1}^K \log f(\mathbf{x}^{(m),c} \theta)$ where $\mathbf{x}^{(m),c} \sim f(\mathbf{x} \mathbf{y}, \theta^{(m)})$
SAEM	Uses a recursive update combining current and past estimates.	$Q(\theta \theta^{(m+1)}) = Q(\theta \theta^{(m)}) + \gamma_k [\log f(\mathbf{x}^{(m)} \theta) - Q(\theta \theta^{(m)})]$ $\mathbf{x}^{(m)} \sim f(\mathbf{x} \mathbf{y}, \theta^{(m)}), \quad 0 < \gamma_k < 1$

TABLE I

COMPARISON OF E-STEP FORMULATIONS IN EM VARIANTS

B. Intractable M-step

In the case of the MAP, the E-step of the EM algorithm remains the same (the expectation is w.r.t. the latent variable only), whereas the M-step becomes

$$\theta_{\text{MAP}}^{(k+1)} = \underset{\theta \in \Omega_\theta}{\operatorname{argmax}} Q(\theta | \theta^{(m)}) + \log f(\theta)$$

which $f(\theta)$ denotes the prior information. Due to the presence of the *a priori*, the M-step might be difficult to compute. We briefly review some variants of the EM algorithm, c.f., Table. II. The key difference among these lies in how the M-step is carried out. In classical EM, the M-step involves a full joint maximization over all parameters, which can become computationally expensive or even intractable for complex models. The ECM (Expectation Conditional Maximization) algorithm addresses this by decomposing the M-step into a series of simpler conditional maximizations over parameter blocks. ECME (Expectation Conditional Maximization Either) extends this idea by replacing some of these conditional steps with direct maximization of the observed-data log-likelihood, often resulting in faster convergence. Lastly, the GEM (Generalized EM) algorithm relaxes the requirement of full maximization entirely, requiring only that the surrogate function increases at each iteration. This flexibility allows for the use of partial updates or gradient-based methods when exact optimization is not feasible. These variants are particularly valuable in high-dimensional, constrained, or non-convex problems where the classical EM algorithm may not be practical.

	M-Step Description	Analytical Equation
EM	Maximizes the full expected complete-data log-likelihood with respect to all parameters.	$\theta^{(m+1)} = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathbb{E}_{\mathbf{x} \sim f(\mathbf{x} \mathbf{y}, \theta^{(m)})} [\log f(\mathbf{x} \theta)]$
ECM	Replaces the full M-step by a sequence of conditional maximizations over subsets of parameters.	For each block b : $\theta_b^{(m+1)} = \underset{\theta_b \in \Theta_b}{\operatorname{argmax}} \mathbb{E}_{\mathbf{x} \sim f(\mathbf{x} \mathbf{y}, \theta_b^{(m)}, \theta_{\bar{b}}^{(m)})} [\log f(\mathbf{x} \theta_b, \theta_{\bar{b}})]$
ECME	Maximizes some parameters, θ_{OL} , using the observed data log-likelihood directly.	$\theta_{\text{OL}}^{(m+1)} = \underset{\theta_{\text{OL}} \in \Theta_{\text{OL}}}{\operatorname{argmax}} \log f(\mathbf{y} \theta_{\text{OL}}, \theta_{\text{OL}}^{(m)})$ and $\theta_{\text{OL}}^{(m+1)} = \underset{\theta_{\text{OL}} \in \Theta_{\text{OL}}}{\operatorname{argmax}} \mathbb{E}_{\mathbf{x} \sim f(\mathbf{x} \mathbf{y}, \theta_{\text{OL}}^{(m)}, \theta_{\text{OL}}^{(m)})} [\log f(\mathbf{x} \theta_{\text{OL}}, \theta_{\text{OL}}^{(m)})]$
GEM	Only requires that the Q -function increases at each iteration, not necessarily maximized.	Find $\theta^{(m+1)}$ such that $Q(\theta^{(m+1)} \theta^{(m)}) \geq Q(\theta^{(m)} \theta^{(m)})$

TABLE II

COMPARISON OF M-STEP FORMULATIONS IN EM VARIANTS

C. Variational EM

blabla

V. VARIATIONAL AUTOENCODERS

The EM algorithm has long been a central tool for estimating parameters in probabilistic models with latent variables. While effective for many structured problems, classical EM and its stochastic variants encounter significant limitations in modern machine learning applications due to scalability issues. Namely, even stochastic versions of EM, which rely on sampling methods to approximate expectations, can struggle to scale efficiently on large datasets or deep generative models.

To overcome these challenges, variational inference provides a powerful alternative. It replaces exact posterior computation with an approximation drawn from a family of tractable distributions, optimized via techniques such as the Evidence Lower Bound (ELBO). Variational Autoencoders (VAEs) integrate this variational inference approach with neural networks, enabling the efficient learning of complex generative models while maintaining scalability and flexibility. In essence, while EM relies on iteratively computing or sampling from intractable posteriors, VAEs use inference via learned encoder networks, making them well-suited for high-dimensional and large-scale problems where classical EM fails.

A. Deep latent variable model

As explained above latent variables, denoted by z , can be incorporated into the model's structure (but are not directly observed in the available data). When performing unconditional modeling of an observed variable y , the corresponding directed graphical model describes a joint probability distribution $p(y, z|\theta)$ involving both the observed data and the hidden variables. To obtain the distribution over the observed variable alone, one marginalizes over the latent variables, resulting

$$p^*(y) \approx p(y|\theta) = \int p(y, z|\theta) dz \quad (34)$$

when $p^*(y)$ is complex, modeling $p(y|\theta)$ tends to be complex too leading to intractable EM algorithm. In this case, it is common to assume a deep latent variable model (DLVM) distributions. The latter parameterize the distributions involved in $p(y, z|\theta) = p(z|\theta)p(y|z, \theta)$ by neural networks.

In order to obtain a tractable inference and learning procedure in the DLVM, we assume a parametric inference model $q(\mathbf{z}|\mathbf{y}, \phi) \approx p(\mathbf{z}|\mathbf{y}, \theta)$. The latter approximation of the posterior, is referred as the *encoder* and ϕ represents the *variational parameters* include weights and biases of the neural network, e.g.,

$$\begin{aligned}\boldsymbol{\mu}, \boldsymbol{\Sigma} &= \text{ENN}(\mathbf{y}, \phi) \\ q(\mathbf{z}|\mathbf{y}, \phi) &= \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\end{aligned}$$

in which ENN refer to the Encoder Neural Network.

B. Linking the MLE to the ELBO

The ELBO is commonly chosen as an objective in the DLVM context. Specifically, the ELBO reads

$$\begin{aligned}(\text{ELBO}) \quad \mathcal{E}_{\phi, \theta}(\mathbf{y}) &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{q(\mathbf{z}|\mathbf{y}, \phi)} \right\} \\ &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \log p(\mathbf{z}, \mathbf{y}|\theta) - \log q(\mathbf{z}|\mathbf{y}, \phi) \}\end{aligned} \quad (35)$$

The ELBO is directly related to the log likelihood, namely,

$$\begin{aligned}\log p(\mathbf{y}|\theta) &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \log p(\mathbf{y}|\theta) \} = E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{p(\mathbf{z}|\mathbf{y}, \theta)} \right\} \\ &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{q(\mathbf{z}|\mathbf{y}, \phi)} \frac{q(\mathbf{z}|\mathbf{y}, \phi)}{p(\mathbf{z}|\mathbf{y}, \theta)} \right\} \\ &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{q(\mathbf{z}|\mathbf{y}, \phi)} \right\} + E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{q(\mathbf{z}|\mathbf{y}, \phi)}{p(\mathbf{z}|\mathbf{y}, \theta)} \right\}\end{aligned} \quad (36)$$

in which the first term denotes the ELBO and reads $E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{p(\mathbf{z}, \mathbf{y}|\theta)}{q(\mathbf{z}|\mathbf{y}, \phi)} \right\}$ and the second term represents the KLD, and reads

$$\text{KLD}(q(\mathbf{z}|\mathbf{y}, \phi) || p(\mathbf{z}|\mathbf{y}, \theta)) = E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \left\{ \log \frac{q(\mathbf{z}|\mathbf{y}, \phi)}{p(\mathbf{z}|\mathbf{y}, \theta)} \right\}.$$

Since the KLD is non-negative, the ELBO is a lower bound of the log likelihood, i.e.,

$$\begin{aligned}\mathcal{E}_{\phi, \theta}(\mathbf{y}) &= \log p(\mathbf{y}|\theta) - \text{KLD}(q(\mathbf{z}|\mathbf{y}, \phi) || p(\mathbf{z}|\mathbf{y}, \theta)) \\ &\leq \log p(\mathbf{y}|\theta)\end{aligned} \quad (37)$$

and $\text{KLD}(q(\mathbf{z}|\mathbf{y}, \phi) || p(\mathbf{z}|\mathbf{y}, \theta))$ can be seen as the distance between the ELBO and the marginal log likelihood and is referred as the *tightness* of the bound. In addition, maximization of the ELBO leads to two important points, 1) minimizes the $\text{KLD}(q(\mathbf{z}|\mathbf{y}, \phi) || p(\mathbf{z}|\mathbf{y}, \theta))$ leading to $q(\mathbf{z}|\mathbf{y}, \phi)$ as close as possible to $p(\mathbf{z}|\mathbf{y}, \theta)$ and 2) it maximizes the marginal likelihood $p(\mathbf{y}|\theta)$. This leads to a better generative model.

C. Linking the EM to the ELBO

The EM algorithm can be seen as *coordinate ascent* on the ELBO $\mathcal{E}_{\phi, \theta}(\mathbf{y})$. Specifically, from (37), consider first maximizing the ELBO for a fixed $\theta^{(m)}$ and at the m -th iteration, i.e.,

$$q^{(m)}(\mathbf{z}|\mathbf{y}, \phi) = \operatorname{argmax}_{q(\mathbf{z}|\mathbf{y}, \phi)} \mathcal{E}_{\phi, \theta^{(m)}}(\mathbf{y}) = \operatorname{argmin}_{q(\mathbf{z}|\mathbf{y}, \phi)} \text{KLD}(q(\mathbf{z}|\mathbf{y}, \phi) || p(\mathbf{z}|\mathbf{y}, \theta^{(m)})) = p(\mathbf{z}|\mathbf{y}, \theta^{(m)})$$

Second, we will optimize over θ with previously estimated $q^{(m)}(\mathbf{z}|\mathbf{y}, \phi) = p(\mathbf{z}|\mathbf{y}, \theta^{(m)})$. Then, using the ELBO expression in (35)

$$\begin{aligned} \theta^{(m+1)} &= \operatorname{argmax}_{\theta \in \Theta} E_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}, \theta^{(m)})} \left\{ \log p(\mathbf{z}, \mathbf{y}|\theta) - \log p(\mathbf{z}|\mathbf{y}, \theta^{(m)}) \right\} \\ &= \operatorname{argmax}_{\theta \in \Theta} E_{\mathbf{z} \sim p(\mathbf{z}|\mathbf{y}, \theta^{(m)})} \{ \log p(\mathbf{z}, \mathbf{y}|\theta) \}. \end{aligned}$$

Consequently, we can notice that for the particular choice of $q^{(m)}(\mathbf{z}|\mathbf{y}, \phi) = p(\mathbf{z}|\mathbf{y}, \theta^{(m)})$, one have the EM algorithm presented in (15) with the complete data given by $\mathbf{x} = \{\mathbf{y}, \mathbf{z}\}$.

D. Maximizing the ELBO through the SGD

In the following, we briefly see how we can make use of the SGD to optimize the ELBO jointly w.r.t. θ and ϕ (this is needed in the case of intractable ELBO or its gradient, which is commonly the case). The main idea is to start with random initial values of θ and ϕ and stochastically optimize their values until convergence. In the case of i.i.d. samples, the ELBO is the average of the individual ELBOs, i.e.,

$$\mathcal{E}_{\phi, \theta}(\mathbf{y}) = \sum_{i=1}^n \mathcal{E}_{\phi, \theta}(\mathbf{y}_i) \quad (38)$$

The individual ELBO and its gradient is generally intractable. We use the SGD as unbiased estimators of $\nabla_{\theta, \phi} \mathcal{E}_{\phi, \theta}(\mathbf{y})$ as explained in (13).

1) *Deriving $\nabla_{\theta} \mathcal{E}_{\phi, \theta}(\mathbf{y})$* : Specifically, we use simple Monte Carlo estimator in order to get unbiased gradients of the ELBO, namely,

$$\begin{aligned} \nabla_{\theta} \mathcal{E}_{\phi, \theta}(\mathbf{y}) &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \nabla_{\theta} (\log p(\mathbf{z}, \mathbf{y}|\theta) - \log q(\mathbf{z}|\mathbf{y}, \phi)) \} \\ &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \nabla_{\theta} \log p(\mathbf{z}, \mathbf{y}|\theta) \} \approx \nabla_{\theta} \log p(\tilde{\mathbf{z}}, \mathbf{y}|\theta) \end{aligned}$$

in which $\tilde{\mathbf{z}} \sim q(\mathbf{z}|\mathbf{y}, \phi)$.

2) *Deriving $\nabla_{\phi} \mathcal{E}_{\phi, \theta}(\mathbf{y})$ through the reparameterization trick:* Computing the gradient w.r.t. ϕ this is more complicated since the expectation depends on ϕ , i.e.,

$$\nabla_{\phi} E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \log q(\mathbf{z}|\mathbf{y}, \phi) \} \neq E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}, \phi)} \{ \nabla_{\phi} \log q(\mathbf{z}|\mathbf{y}, \phi) \}.$$

But the so-called *reparameterization trick*, which can be assimilated as a change of variables, is an easy way to overcome this issue. Namely, let us link $\mathbf{z}$ with an other random variable $\mathbf{e}$, as,

$$\mathbf{z} = \mathbf{r}(\mathbf{e}, \mathbf{y}, \phi)$$

with $\mathbf{r}(\cdot)$ is differentiable and invertible and $\mathbf{e} \sim p(\mathbf{e})$ is independent of $\mathbf{y}$. Consequently,

$$\begin{aligned} \nabla_{\phi} \mathcal{E}_{\phi, \theta}(\mathbf{y}) &= \nabla_{\phi} E_{\mathbf{e} \sim p(\mathbf{e})} \{ \log p(\mathbf{z}, \mathbf{y}|\theta) - \log q(\mathbf{z}|\mathbf{y}, \phi) \} \\ &\approx \nabla_{\phi} \log p(\tilde{\mathbf{z}}, \mathbf{y}|\theta) - \log q(\tilde{\mathbf{z}}|\mathbf{y}, \phi) \end{aligned}$$

in which, $\tilde{\mathbf{e}} \sim p(\mathbf{e})$ and then, $\tilde{\mathbf{z}} = \mathbf{r}(\tilde{\mathbf{e}}, \mathbf{y}, \phi)$.

Example 5.1 (Computation of unbiased estimate of ELBO with Gaussian inference model): In the following, we consider

- The Gaussian inference model writes $q(\mathbf{z}|\mathbf{y}, \phi) = \mathcal{N}(\bar{\boldsymbol{\mu}}(\phi, \mathbf{x}), \bar{\boldsymbol{\Sigma}}(\phi, \mathbf{x}))$ which aims at approximating the true $p(\mathbf{z}|\mathbf{y})$, such that the mean and covariance will be learned using an encoder neural network of parameter ϕ and input $\mathbf{y}$, $\text{ENN}(\phi, \mathbf{y})$.
- The generative model reads $p(\mathbf{z}, \mathbf{y}|\theta) = p(\mathbf{y}|\mathbf{z}, \theta)p(\mathbf{z})$ with $p(\mathbf{y}|\mathbf{z}, \theta) = \mathcal{N}(\boldsymbol{\mu}(\theta, \mathbf{z}), \boldsymbol{\Sigma}(\theta, \mathbf{y}))$, such that the mean and covariance will be learned using an decoder neural network of parameter θ and input $\mathbf{z}$, $\text{DNN}(\phi, \mathbf{y})$ and $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

From $q(\mathbf{z}|\mathbf{y}, \phi) = \mathcal{N}(\bar{\boldsymbol{\mu}}(\phi, \mathbf{y}), \bar{\boldsymbol{\Sigma}}(\phi, \mathbf{y}))$, the reparameterization reads

$$\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{39}$$

$$\mathbf{z} = \bar{\boldsymbol{\mu}}(\phi, \mathbf{y}) + \bar{\mathbf{L}}(\phi, \mathbf{y})\mathbf{e}$$

This parametrization corresponds to the Cholesky decomposition of the covariance matrix, i.e., $\bar{\boldsymbol{\Sigma}}(\phi, \mathbf{y}) = \bar{\mathbf{L}}(\phi, \mathbf{y})\bar{\mathbf{L}}(\phi, \mathbf{y})^T$. In order to impose a Cholesky form for the matrix $\bar{\mathbf{L}}(\phi, \mathbf{y})$ we can proceed as

$$\begin{aligned} \bar{\boldsymbol{\mu}}(\phi, \mathbf{y}), \bar{\boldsymbol{\sigma}}(\phi, \mathbf{y}), \bar{\mathbf{L}}'(\phi, \mathbf{y}) &\leftarrow \text{ENN}(\phi, \mathbf{y}) \\ \bar{\mathbf{L}}(\phi, \mathbf{y}) &\leftarrow \mathbf{L}_{\text{mask}} \odot \bar{\mathbf{L}}'(\phi, \mathbf{y}) + \text{diag}(\bar{\boldsymbol{\sigma}}(\phi, \mathbf{y})) \end{aligned}$$

in which $\leftarrow L_{\text{mask}}$ contains zeros on and above the diagonal. and then, we can sample $\mathbf{z}$ from (39).

From (35) and (38), the individual ELBO writes

$$\mathcal{E}_{\phi, \theta}(\mathbf{y}_i) = E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}_i, \phi)} \{ \log p(\mathbf{y}_i|\mathbf{z}, \theta) \} + \log p(\mathbf{z}) - E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{y}_i, \phi)} \{ \log q(\mathbf{z}|\mathbf{y}_i, \phi) \} \quad (40)$$

The ELBO in (40) contains the term $q(\mathbf{z}|\mathbf{y}_i, \phi)$ that must be evaluated. The relation between densities of $\mathbf{e}$ and $\mathbf{z}$, with an invertible function $\mathbf{r}(\cdot)$, reads

$$\begin{aligned} q(\mathbf{z}|\mathbf{y}, \phi) &= \log p(\mathbf{e}) - \log \left| \det \frac{\partial \mathbf{z}}{\partial \mathbf{e}} \right| \\ &= \log \mathcal{N}(\mathbf{0}, \mathbf{I}) - \sum_j \log |[\bar{\mathbf{L}}(\phi, \mathbf{y})]_{jj}| \\ &= -\frac{1}{2} \sum_j ([\mathbf{e}]_j^2 + \log(2\pi) + \log([\boldsymbol{\sigma}]_j)) . \end{aligned} \quad (41)$$

Summing up these led us to the following algorithm

Unbiased estimate of ELBO with Gaussian inference model

Initialization: θ, ϕ

- $\bar{\boldsymbol{\mu}}(\phi, \mathbf{y}_i), \bar{\boldsymbol{\sigma}}(\phi, \mathbf{y}_i), \bar{\mathbf{L}}'(\phi, \mathbf{y}_i) \leftarrow \text{ENN}(\phi, \mathbf{y}_i)$
 - $\bar{\mathbf{L}}(\phi, \mathbf{y}_i) \leftarrow L_{\text{mask}} \odot \bar{\mathbf{L}}'(\phi, \mathbf{y}_i) + \text{diag}(\bar{\boldsymbol{\sigma}}(\phi, \mathbf{y}_i))$
 - $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - $\mathbf{z} \leftarrow \bar{\boldsymbol{\mu}}(\phi, \mathbf{y}_i) + \bar{\mathbf{L}}(\phi, \mathbf{y}_i) \mathbf{e}$
 - $\boldsymbol{\mu}(\theta, \mathbf{z}_i), \boldsymbol{\Sigma}(\theta, \mathbf{y}_i) \leftarrow \text{DNN}(\theta, \mathbf{z})$
 - $(p(\mathbf{z})) \tilde{\mathcal{E}}_{\mathbf{z}} = \log \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$
 - $(\log q(\mathbf{z}|\mathbf{y}_i, \phi) \text{ approximation}) \tilde{\mathcal{E}}_{\mathbf{z}|\mathbf{y}_i, \phi} = -\frac{1}{2} \sum_j ([\mathbf{e}]_j^2 + \log(2\pi) + \log([\boldsymbol{\sigma}]_j)) .$
 - $(\log p(\mathbf{y}_i|\mathbf{z}, \theta) \text{ approximation}) \tilde{\mathcal{E}}_{\mathbf{y}_i|\mathbf{z}, \theta} = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}(\theta, \mathbf{z}_i), \boldsymbol{\Sigma}(\theta, \mathbf{y}_i))$
 - *(One individual ELBO approximation)* $\tilde{\mathcal{E}}_{\phi, \theta}(\mathbf{y}_i) = \tilde{\mathcal{E}}_{\mathbf{y}_i|\mathbf{z}, \theta} + \tilde{\mathcal{E}}_{\mathbf{z}} - \tilde{\mathcal{E}}_{\mathbf{z}|\mathbf{y}_i, \phi}$
-

VI. APPENDIX

A. Other examples

1) *Robust mean estimation*: In this Section, we present a simple example of a robust mean estimation [6], [7]. The extension to the case of DoA estimation can be found in, e.g., Maximum likelihood estimation of compound-Gaussian clutter and target parameters *Wang et al. IEEE TSP 2016*.

Let us consider the following model

$$\mathbf{y}_t = \boldsymbol{\mu} + \mathbf{e}_t \quad (42)$$

in which $\mathbf{y}_t \in \mathbb{R}^P$ and we consider a compound-Gaussian noise modeling given as

$$\mathbf{e}_t = \frac{1}{\sqrt{\tau_t}} \mathbf{n}_t$$

where $\tau_t \sim p_\tau(\nu)$ in which ν is a deterministic parameter, and $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, in which the covariance matrix $\boldsymbol{\Sigma}$ is unknown (in the case of direction of arrival estimation, one can replace $\boldsymbol{\mu}$ by $\mathbf{A}(\boldsymbol{\theta})\mathbf{s}_t$ and consider the complex counterpart of the noise model). Then, the DoA, $\boldsymbol{\theta}$ can be deduced from the one to one mapping between $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ or by using the SAGE algorithm described below [8], [9].

From (42), one obtain

$$\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu, \tau \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\tau} \boldsymbol{\Sigma}) \quad (43)$$

In the case of a student-t distributed data, we have

$$\tau|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu \sim \text{Gamma}(\frac{\nu}{2}, \frac{\nu}{2})$$

where the $\text{Gamma}(\alpha, \beta)$ density function reads, for $\tau > 0, \beta > 0, \alpha > 0$,

$$\text{Gamma}(\frac{\nu}{2}, \frac{\nu}{2}) = \frac{\beta^\alpha \tau^{\alpha-1} \exp(-\beta\tau)}{\Gamma(\alpha)}$$

Note that for $\nu \rightarrow \infty$, the $\tau \rightarrow 1$ with probability one, thus, $\mathbf{y}$ becomes marginally normal, i.e., $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. And the marginal distribution reads

$$f(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma(\frac{\nu+P}{2})|\boldsymbol{\Sigma}|^{-\frac{1}{2}}}{(\pi\nu)^{\frac{P}{2}}\Gamma(\frac{\nu}{2})\left(1 + \frac{d_{\mathbf{y}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\nu}\right)^{\frac{\nu+P}{2}}} \quad (44)$$

in which the Mahalanobis distance from $\mathbf{y}$ to the center $\boldsymbol{\mu}$ with respect to $\boldsymbol{\Sigma}$ reads

$$d_{\mathbf{y}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})$$

Note that the above distribution is ellipsoidally symmetric about the mean $\boldsymbol{\mu}$ since the density is the same for all $\mathbf{y}$ that have the same $\boldsymbol{\Sigma}$ distance from $\boldsymbol{\mu}$.

Before we derive the ML and the EM algorithm for $\theta = [\{\boldsymbol{\mu}\}, \{\boldsymbol{\Sigma}\}, \nu]$ we first recall some important results that can be found in *Anderson 1985, p.35*: let us assume that $\mathbf{x}$ and $\mathbf{z}$ are jointly Gaussian, s.t., $[\mathbf{x}^T, \mathbf{y}^T]^T \sim \mathcal{N}(\boldsymbol{\mu}, \frac{\boldsymbol{\Sigma}}{\tau})$, thus

$$\begin{aligned} \mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu, \tau &\sim \mathcal{N}(\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X/\tau) \\ \mathbf{y}|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu, \tau &\sim \mathcal{N}(\boldsymbol{\mu}_{Y|X}, \boldsymbol{\Sigma}_{Y|X}^*/\tau) \\ \boldsymbol{\mu}_{Y|X} &= \boldsymbol{\mu}_Y - \boldsymbol{\Sigma}_{Y,X} \boldsymbol{\Sigma}_X^{-1} (\mathbf{x} - \boldsymbol{\mu}_X) \\ \boldsymbol{\Sigma}_{Y|X}^* &= \boldsymbol{\Sigma}_Y - \boldsymbol{\Sigma}_{Y,X} \boldsymbol{\Sigma}_X^{-1} \boldsymbol{\Sigma}_{X,Y} \end{aligned} \tag{45}$$

a) *ML of θ with observed $\mathbf{y}$ and τ* : Let us consider n i.i.d. observation

$$\mathbf{y}_t|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \tau_t \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}/\tau_t)$$

in which

$$\tau_t \sim \text{Gamma}(\nu/2, \nu/2)$$

From above, it is natural to chose the complete data as $\{\mathbf{y}_1, \dots, \mathbf{y}_n, \tau_1, \dots, \tau_n\}$ mainly because in this case the ML of the complete data reduces to the classical Gaussian ML for the mean and the covariance matrix.

It is worth mentioning that the log likelihood can be factorized in this case, specifically,

$$\log p(\mathbf{y}_1, \dots, \mathbf{y}_n, \tau_1, \dots, \tau_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \log p(\mathbf{y}_1, \dots, \mathbf{y}_n, \tau_1, \dots, \tau_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(\tau_1, \dots, \tau_n|\nu)$$

in which

$$\begin{aligned} \log p(\mathbf{y}_1, \dots, \mathbf{y}_n, \tau_1, \dots, \tau_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= -n/2 \log |\boldsymbol{\Sigma}| - 1/2 \text{Tr}(\boldsymbol{\Sigma}^{-1}) \sum_{i=1}^n \tau_i \mathbf{y}_i \mathbf{y}_i^T \\ &\quad + \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \sum_{i=1}^n \tau_i \mathbf{y}_i - \frac{1}{2} \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \sum_{i=1}^n \tau_i \end{aligned}$$

and

$$\log p(\tau_1, \dots, \tau_n|\nu) = -n \log(\Gamma(\nu/2)) + \frac{n\nu}{2} \log(\nu/2) + \frac{\nu}{2} \sum_{i=1}^n (\log(\tau_i) - \tau_i)$$

from which one deduce easily the sufficient statistics w.r.t. the complete data

$$S_{\tau, \mathbf{y}} = \sum_{i=1}^n \tau_i \mathbf{y}_i \quad (46)$$

$$S_{\tau, \mathbf{y}, \mathbf{y}} = \sum_{i=1}^n \tau_i \mathbf{y}_i \mathbf{y}_i^T \quad (47)$$

$$S_{\tau} = \sum_{i=1}^n \tau_i \quad (48)$$

$$S_{\tau, \tau} = \sum_{i=1}^n (\log(\tau_i) - \tau_i) \quad (49)$$

Given these sufficient statistics, the MLE reads

$$\hat{\boldsymbol{\mu}} = \frac{S_{\tau, \mathbf{y}}}{S_{\tau}} = \frac{\sum_{i=1}^n \tau_i \mathbf{y}_i}{\sum_{i=1}^n \tau_i} \quad (50)$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} \left(S_{\tau, \mathbf{y}, \mathbf{y}} - \frac{1}{S_{\tau}} S_{\tau, \mathbf{y}} S_{\tau, \mathbf{y}}^T \right) = \frac{1}{n} \sum_{i=1}^n w_i^{(m+1)} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}) (\mathbf{y}_i - \hat{\boldsymbol{\mu}})^T \quad (51)$$

Regarding ν , no closed form expression is available and one has to solve the following

$$\frac{\partial \log \Gamma(\nu/2)}{\partial \nu} + \log(\nu/2) + \frac{1}{n} S_{\tau, \tau} + 1 = 0 \quad (52)$$

b) EM of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with ν known: From above the E-step of the EM algorithm, i.e., the computation of the Q function, reduces to find the conditional expectation of the complete-data sufficient statistics $S_{\tau, \mathbf{y}}, S_{\tau, \mathbf{y}, \mathbf{y}}, S_{\tau}$ and $S_{\tau, \tau}$ given the observed values of $\{\mathbf{y}\}$, the known value of ν and the current estimate of $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$. The M-step involves weighted least squares estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as in (50) and (51). Let us denote $\boldsymbol{\Omega}^{(m)} = \{\mathbf{y}_1, \dots, \mathbf{y}_n, \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}, \nu\}$. Consequently³,

$$w_i^{(m+1)} = E_{\tau|\boldsymbol{\Omega}^{(m)}} \{\tau_i\} = \frac{\nu + p}{\nu + \delta_i^{(m)}} \quad (53)$$

³This can be obtained noticing the fact that $p(\tau|\mathbf{y}, \boldsymbol{\Sigma}, \boldsymbol{\mu}, \nu) \propto p(\mathbf{y}|\tau, \boldsymbol{\Sigma}, \boldsymbol{\mu}, \nu)p(\tau|\boldsymbol{\Sigma}, \boldsymbol{\mu}, \nu)$ leading to $\tau|\mathbf{y}, \boldsymbol{\Sigma}, \boldsymbol{\mu}, \nu = \tau|\delta_i, \nu \sim \text{Gamma}(\frac{\nu+p}{2}, \frac{\nu+\delta_i}{2})$

in which $\delta_i^{(m)} = (\mathbf{y}_i - \boldsymbol{\mu}^{(m)})^T (\boldsymbol{\Sigma}^{(m)})^{-1} (\mathbf{y}_i - \boldsymbol{\mu}^{(m)})$. Finally, using the EM formulation for missing data, we need⁴

$$E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau, \mathbf{y}}\} = \sum_{i=1}^n w_i^{(m+1)} \mathbf{y}_i \quad (54)$$

$$E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau, \mathbf{y}, \mathbf{y}}\} = \sum_{i=1}^n w_i^{(m+1)} \mathbf{y}_i \mathbf{y}_i^T \quad (55)$$

$$E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau}\} = \sum_{i=1}^n w_i^{(m+1)} \quad (56)$$

Plugging the above in $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = \sum_{i=1}^n E_{\tau_i|\mathbf{y}_i, \boldsymbol{\theta}^{(m)}, \nu} \{\log p(\tau_i, \mathbf{y}_i|\boldsymbol{\theta}, \nu)\}$ in which $\boldsymbol{\theta} = \{\boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}\}$, the M-step reduces to (see Section. VI-A1a)

$$\hat{\boldsymbol{\mu}}^{(m+1)} = \frac{S_{\tau, \mathbf{y}}}{S_{\tau}} = \frac{\sum_{i=1}^n w_i^{(m+1)} \mathbf{y}_i}{\sum_{i=1}^n w_i^{(m+1)}} \quad (57)$$

$$\hat{\boldsymbol{\Sigma}}^{(m+1)} = \sum_{i=1}^n \frac{1}{n} w_i^{(m+1)} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}) (\mathbf{y}_i - \hat{\boldsymbol{\mu}})^T \quad (58)$$

c) *EM of ν , $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$:* In the case of unknown ν , one has to first to compute⁵

$$E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau, \tau}\} = \sum_{i=1}^n \left(\phi\left(\frac{p + \nu^{(m)}}{2}\right) - \log\left(\frac{p + \nu^{(m)}}{2}\right) \right) + \sum_{i=1}^n \left(\log w_i^{(m+1)} - w_i^{(m+1)} \right) \quad (59)$$

in which $\phi(\cdot)$ denotes the digamma function, $\boldsymbol{\Omega}^{(m)} = \{\mathbf{y}_1, \dots, \mathbf{y}_n, \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}, \nu^{(m)}\}$ and as above

$$w_i^{(m+1)} = E_{\tau|\boldsymbol{\Omega}^{(m)}} \{\tau_i\} = \frac{\nu^{(m)} + p}{\nu^{(m)} + \delta_i^{(m)}}$$

Regarding the M-step, we notice that there is a separation of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and ν . Consequently, the M-step of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ remains the same (with $\nu^{(m)}$ instead of ν). The M-step regarding ν is more difficult and one has to solve the following

$$-\phi(\nu/2) + \log(\nu/2) + 1 + \sum_{i=1}^n \left(\phi\left(\frac{p + \nu^{(m)}}{2}\right) - \log\left(\frac{p + \nu^{(m)}}{2}\right) \right) + \sum_{i=1}^n \left(\log w_i^{(m+1)} - w_i^{(m+1)} \right) \quad (60)$$

2) *Robust direction of arrival estimation: The one source case:* Most of array processing technique are designed under the assumption of Gaussian noise. Nevertheless, this assumption might be unrealistic leading to a dramatic performance loss. The non Gaussian environment rises, for example, in the presence

⁴ $E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau, \tau}\}$ is needed here since ν is assumed to be known and the M-step does not depend on $E_{\tau|\boldsymbol{\Omega}^{(m)}} \{S_{\tau, \tau}\}$

⁵ Using the fact that if a random variable R has a Gamma(α, β) distribution then, $E\{\log R\} = \phi(\alpha) - \log \beta$ where $\phi(\cdot)$ denotes the Digamma function (see McLachlan and Krishnan 1997, section 2.6, 5.8, eq 4.152).

of outliers, an heterogenous environment or in the recent high resolution radars for which the central limit theorem is not valid any more and thus the radar clutter cannot be correctly modeled as a Gaussian process. To overcome this limitation, one alternative is to consider a Gaussian mixture. Its flexibility allows to overcome the performance loss due to a Gaussian assumption.

In the following, we consider one far field source impinging on a linear array. In this context, the observation (array output) $\mathbf{Y} = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$, in which each snapshots reads

$$\mathbf{y}_i = \mathbf{a}(\psi)s_i + \mathbf{n}_i \quad (61)$$

in which ψ is the direction of arrival, $\mathbf{n}_i$ represents the noise component model constituted of a mixture of k Gaussians with zero mean and unknown covariance matrices Σ_l for $l = 1, \dots, k$. The number of mixtures should be estimated, nevertheless, it has been noticed from numerical simulations that $k = 4$ mixture terms provides very good detection and estimation performance in a variety of noise pdf cases, including stable distributions.

Let us define the unknown parameter as

$$\boldsymbol{\theta} = [\psi, s_1, \dots, s_n, w_1, \dots, w_k, \text{vec}(\Sigma_1)^T, \dots, \text{vec}(\Sigma_k)^T]^T$$

in which the weights, for each mixture, satisfy $w_j > 0$ and $\sum_{j=1}^k w_j = 1$. Consequently, with some abuse of notation, we can write

$$p(Y_i = \mathbf{y}_i | \boldsymbol{\theta}) = \sum_{j=1}^k w_j \mathcal{CN}(\boldsymbol{\eta}_i, \Sigma_j)$$

in which $\boldsymbol{\eta}_i = \mathbf{a}(\psi)s_i$. Let us denote $\gamma_{ij}^{(m)}$ the estimate of the probability that the i -th sample was generated by the j -th Gaussian component at the m -th iteration. Thus,

$$\gamma_{ij}^{(m)} \triangleq P(Z_i = j | \mathbf{y}_i, \boldsymbol{\theta}^{(m)}) = \frac{w_j^{(m)} \mathcal{CN}(\boldsymbol{\eta}_i^{(m)}, \Sigma_j^{(m)})}{\sum_{l=1}^k w_j^{(l)} \mathcal{CN}(\boldsymbol{\eta}_i^{(l)}, \Sigma_j^{(l)})} \quad (62)$$

which clearly satisfies $\sum_{l=1}^k \gamma_{il}^{(m)} = 1$.

Since the samples are i.i.d., the E-step reduces to the calculation of $Q_i(\boldsymbol{\theta} | \boldsymbol{\theta}^{(m)})$ which reads

$$\begin{aligned}
Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{Z_i|\mathbf{y}_i, \boldsymbol{\theta}^{(m)}} \{\log p(\mathbf{y}_i, Z_i|\boldsymbol{\theta})\} \\
&= \sum_{j=1}^k P(Z_i = j|\mathbf{y}_i, \boldsymbol{\theta}^{(m)}) \log p(\mathbf{y}_i, Z_i = j|\boldsymbol{\theta}) \\
&= \sum_{j=1}^k \gamma_{ij}^{(m)} \log(w_j \mathcal{CN}(\boldsymbol{\eta}_i, \boldsymbol{\Sigma}_j)) \\
&\propto \sum_{j=1}^k \gamma_{ij}^{(m)} (\log w_j - \log |\boldsymbol{\Sigma}_j| - (\mathbf{y}_i - \boldsymbol{\eta}_i)^H \boldsymbol{\Sigma}_j^{-1} (\mathbf{y}_i - \boldsymbol{\eta}_i)) + C
\end{aligned} \tag{63}$$

in which C denotes any values that is independent of $\boldsymbol{\theta}$. Consequently,

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \propto \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^{(m)} (\log w_j - \log |\boldsymbol{\Sigma}_j| - (\mathbf{y}_i - \boldsymbol{\eta}_i)^H \boldsymbol{\Sigma}_j^{-1} (\mathbf{y}_i - \boldsymbol{\eta}_i)) \tag{64}$$

The M-step reduces then to solve

$$\begin{aligned}
\boldsymbol{\theta}^{(m+1)} &= \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \\
\text{subject to } & w_j > 0, \sum_{j=1}^k w_j = 1, \quad j = 1, \dots, k
\end{aligned} \tag{65}$$

$$\boldsymbol{\Sigma}_j \succ 0 \quad j = 1, \dots, k \tag{66}$$

It is noticed that the weight can be solved independently, (using the Lagrange multiplier as in Section .III-D) as

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{l=1}^k n_l^{(m)}}, \quad j = 1, \dots, k \tag{67}$$

where $n_j^{(m)} = \sum_{i=1}^n \gamma_{ij}^{(m)}$ which satisfy (65).

What remains, i.e., the optimal values of $\psi, s_1, \dots, s_n, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_k$, can be solved iteratively by using a block coordinate descent algorithm. Specifically, equating the derivative of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ w.r.t. s_i to zero, leads

$$s_i^{(m+1)} = \frac{\left(\sum_{j=1}^k \gamma_{ij}^{(m)} \left(\boldsymbol{\Sigma}_j^{(m)} \right)^{-1} \mathbf{a}(\psi^{(m)}) \right)^H \mathbf{y}_i}{\left(\sum_{j=1}^k \gamma_{ij}^{(m)} \left(\boldsymbol{\Sigma}_j^{(m)} \right)^{-1} \mathbf{a}(\psi^{(m)}) \right)^H \mathbf{a}(\psi^{(m)})} \tag{68}$$

Regarding the direction of arrival, no closed form expression is available, thus, a Newton type or a classical 1D grid search can be performed

$$\psi^{(m+1)} = \arg \min_{\psi} \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^{(m)} (\mathbf{y}_i - \mathbf{a}(\psi) s_i^{(m+1)})^H (\boldsymbol{\Sigma}_j^{(m)})^{-1} (\mathbf{y}_i - \mathbf{a}(\psi) s_i^{(m+1)}) \quad (69)$$

In the same manner, equating the derivative of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ w.r.t. $\boldsymbol{\Sigma}_j$ to zero, leads

$$\begin{aligned} \frac{Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})}{\partial \boldsymbol{\Sigma}_j} &= -n_j(m) \frac{\partial \log |\boldsymbol{\Sigma}_j|}{\partial \boldsymbol{\Sigma}_j} - \sum_{i=1}^n \gamma_{ij}^{(m)} \frac{\partial (\mathbf{y}_i - \mathbf{a}(\psi) s_i)^H \boldsymbol{\Sigma}_j^{-1} (\mathbf{y}_i - \mathbf{a}(\psi) s_i)}{\partial \boldsymbol{\Sigma}_j} \\ &= n_j(m) \boldsymbol{\Sigma}_j^{-1} + \sum_{i=1}^n \gamma_{ij}^{(m)} \boldsymbol{\Sigma}_j^{-1} (\mathbf{y}_i - \mathbf{a}(\psi) s_i) (\mathbf{y}_i - \mathbf{a}(\psi) s_i)^H \boldsymbol{\Sigma}_j^{-1} \end{aligned}$$

which leads to

$$\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{n_j(m)} \sum_{i=1}^n \gamma_{ij}^{(m)} (\mathbf{y}_i - \mathbf{a}(\psi^{(m+1)}) s_i^{(m+1)}) (\mathbf{y}_i - \mathbf{a}(\psi^{(m+1)}) s_i^{(m+1)})^H \quad (70)$$

Finally, the EM reduces to

EM for robust direction of arrival estimation (the one source case)

- *Initialization:* Chose initial estimates $w_j^{(0)}, \boldsymbol{\eta}_i^{(0)}, \boldsymbol{\Sigma}_j^{(0)}$ for $j = 1, \dots, k$ and $i = 1, \dots, n$
- *E-step:* Compute

$$\gamma_{ij}^{(m)} = \frac{w_j^{(m)} \mathcal{CN}(\boldsymbol{\eta}_i^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^k w_j^{(l)} \mathcal{CN}(\boldsymbol{\eta}_i^{(l)}, \boldsymbol{\Sigma}_j^{(l)})}$$

- *M-step:* Update

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{l=1}^k n_l^{(m)}}, \quad j = 1, \dots, k$$

$$s_i^{(m+1)} = \frac{\left(\sum_{j=1}^k \gamma_{ij}^{(m)} \left(\boldsymbol{\Sigma}_j^{(m)} \right)^{-1} \mathbf{a}(\psi^{(m)}) \right)^H \mathbf{y}_i}{\left(\sum_{j=1}^k \gamma_{ij}^{(m)} \left(\boldsymbol{\Sigma}_j^{(m)} \right)^{-1} \mathbf{a}(\psi^{(m)}) \right)^H \mathbf{a}(\psi^{(m)})}$$

$$\psi^{(m+1)} = \arg \min_{\psi} \sum_{i=1}^n \sum_{j=1}^k \gamma_{ij}^{(m)} (\mathbf{y}_i - \mathbf{a}(\psi) s_i^{(m+1)})^H (\boldsymbol{\Sigma}_j^{(m)})^{-1} (\mathbf{y}_i - \mathbf{a}(\psi) s_i^{(m+1)})$$

$$\boldsymbol{\eta}_i^{(m+1)} = \mathbf{a}(\psi^{(m+1)}) s_i^{(m+1)}$$

$$\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{n_j(m)} \sum_{i=1}^n \gamma_{ij}^{(m)} (\mathbf{y}_i - \mathbf{a}(\psi^{(m+1)}) s_i^{(m+1)}) (\mathbf{y}_i - \mathbf{a}(\psi^{(m+1)}) s_i^{(m+1)})^H$$

- *Convergence check:* Compute $l(\boldsymbol{\theta}^{(m+1)})$, if $|l(\boldsymbol{\theta}^{(m+1)}) - l(\boldsymbol{\theta}^{(m)})| > \delta$, for a chosen δ , return to E-step, otherwise end the algorithm

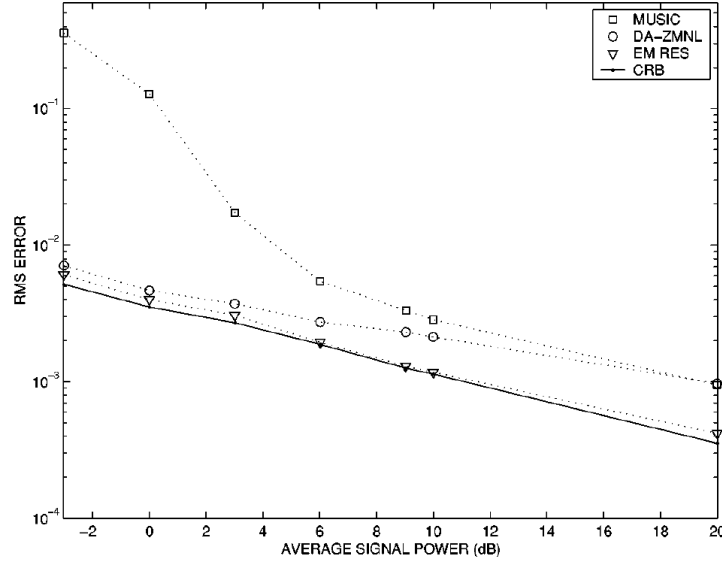


Fig. 1. Simulations with Gaussian mixture noise from *Kozick and Sadler, IEEE TSP, 2010*

Remark: Learning a Gaussian mixture models

Learning GMM (Gaussian mixture models) using EM also referred as EM clustering is detailed in *R. Gupta et al. FTSP 2010*. Here we describe briefly the model setup and we present some results in order to show the usefulness of the EM.

Learning a GMM is a generalization of learning an optimal mixture of fixed (Gaussian) models and can be obtained using exactly the same methodology as above (i.e., the same steps and derivations leading to the EM of the robust direction of arrival estimation scenario). Specifically, one has to estimate the k means and k covariance matrices, and k weights that specify how likely each Gaussian is to be chosen. This set of parameters constitute $\boldsymbol{\theta}$. Consider that we have n samples, such that, each sample $\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\eta}_i, \boldsymbol{\Sigma}_i)$. And we assume that the (real case) GMM has the following density (we some abuse of notation)

$$P(Y_i = \mathbf{y}_i | \boldsymbol{\theta}) = \sum_{j=1}^k w_j \mathcal{N}(\boldsymbol{\eta}_j, \boldsymbol{\Sigma}_j)$$

with $w_i > 0$, $\sum_{j=1}^k w_j = 1$ and $\theta = \{(w_j, \boldsymbol{\eta}_j, \boldsymbol{\Sigma}_j)\}$

The EM reduces to

EM for learning GMM

- *Initialization:* Chose initial estimates $w_j^{(0)}, \boldsymbol{\eta}_j^{(0)}, \boldsymbol{\Sigma}_j^{(0)}$ for $j = 1, \dots, k$
- *E-step:* Compute

$$\gamma_{ij}^{(m)} = \frac{w_j^{(m)} \mathcal{N}(\mathbf{y}_i | \boldsymbol{\eta}_j^{(m)}, \boldsymbol{\Sigma}_j^{(m)})}{\sum_{l=1}^k w_j^{(l)} \mathcal{N}(\mathbf{y}_i | \boldsymbol{\eta}_j^{(l)}, \boldsymbol{\Sigma}_j^{(l)})}$$

- *M-step:* Update

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{l=1}^k n_l^{(m)}}, \quad j = 1, \dots, k$$

$$\boldsymbol{\eta}_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^n \gamma_{ij}^{(m)} \mathbf{y}_i$$

$$\boldsymbol{\Sigma}_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^n \gamma_{ij}^{(m)} (\mathbf{y}_i - \boldsymbol{\eta}_j^{(m+1)})(\mathbf{y}_i - \boldsymbol{\eta}_j^{(m+1)})^T$$

- *Convergence check:* Compute $l(\theta^{(m+1)})$, if $|l(\theta^{(m+1)}) - l(\theta^{(m)})| > \delta$, for a chosen δ , return to E-step, otherwise end the algorithm

3) *Multi-source separation: Parallelization using the EM:* In Section (VI-A2), the mono-source case has been considered. One can notice that extending the above study to the multi-source case is straightforward. Nevertheless, this leads the M-step to be computationally a heavy procedure. Specifically, for m sources, (69) becomes a non convex highly non linear m -D optimization problem.

In this section, we overcome this problem by using the EM algorithm. It is worth mentioning that for sake of simplicity we consider here the case of m sources impinging on an array in the presence of zero mean Gaussian with known covariance matrix and known signal source. The extension to the case of multi-source case with mixture of Gaussian with unknown covariance matrix is given in *Kozick & Sadler IEEE TSP 2000*, but the methodology remains the same.

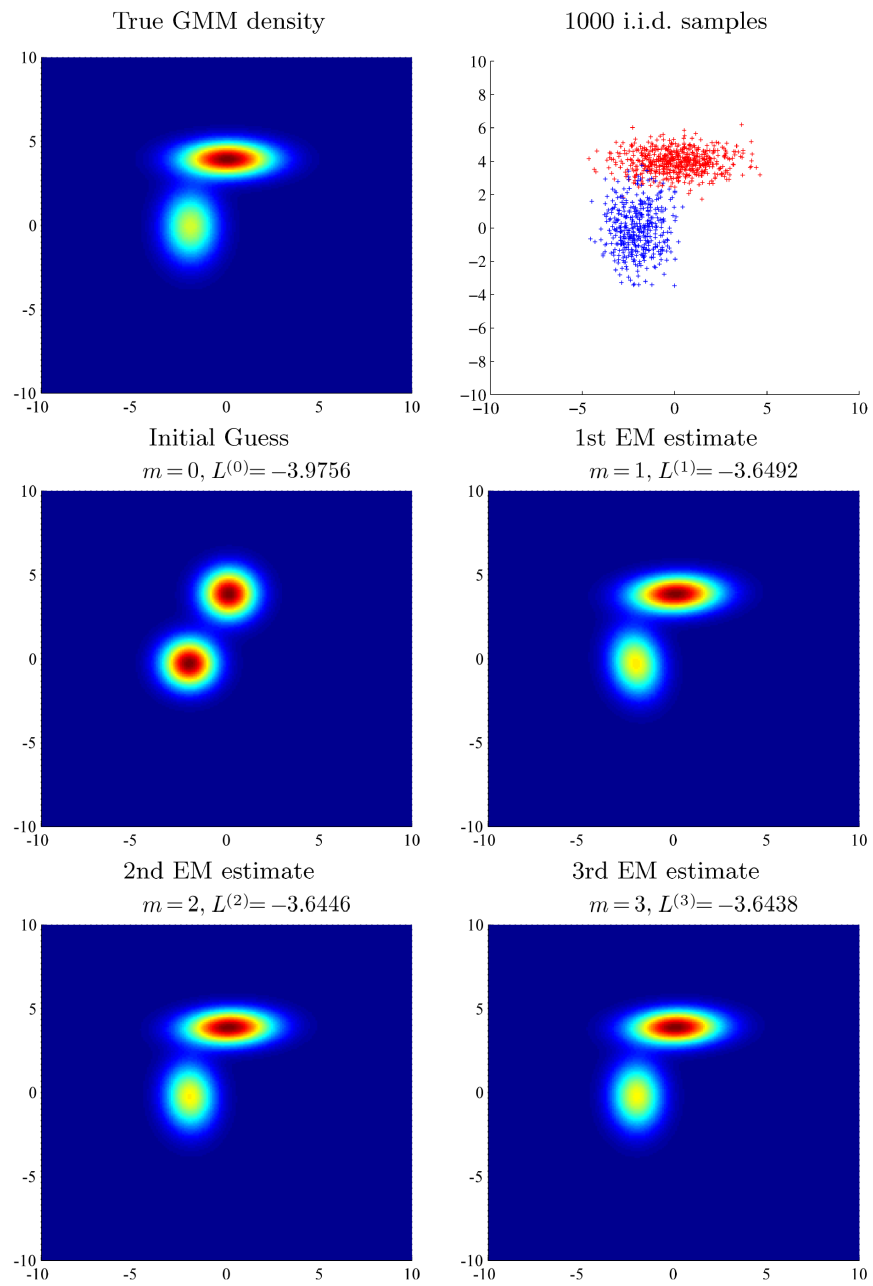


Fig. 2. GMM fitting example from *Gupta et al. FTSP 2010*

Consequently, in the following, we consider the following model

$$\mathbf{y}_i = \sum_{p=1}^m \mathbf{a}(\psi_p) s_{p,i} + \mathbf{n}_i \quad (71)$$

$$= \mathbf{A}(\boldsymbol{\psi}) \mathbf{s}_i + \mathbf{n}_i \quad (72)$$

in which the steering matrix reads $\mathbf{A}(\boldsymbol{\psi}) = [\mathbf{a}(\psi_1), \dots, \mathbf{a}(\psi_m)]$, with $\boldsymbol{\psi} = [\psi_1, \dots, \psi_m]^T$ denotes the m direction of arrivals, the source signal $\mathbf{s}_i = [s_{1,i}, \dots, s_{m,i}]^T$ and $\mathbf{n}_i$ represents a Gaussian additive noise with $\boldsymbol{\Sigma}$ as a covariance matrix. In this case the unknown parameter $\boldsymbol{\theta}$ reduces to $\boldsymbol{\psi}$, i.e., $\boldsymbol{\theta} \triangleq \boldsymbol{\psi}$ and in the same manner we consider that $\theta_p \triangleq \psi_p$ for $p = 1, \dots, m$. Thus, in the following we will consider θ_p instead of ψ_p .

To make use of the EM, we have first to specify the complete data. In our scenario, specifying the complete data is rather straightforward. A proper choice in order to have a reduced complexity step to maximizes $p(\mathbf{x}|\boldsymbol{\theta})$ while maintaining the Markov relationship, i.e., $\boldsymbol{\theta} \rightarrow X \rightarrow Y$, is to consider the following complete data, for the i -th observation and the p -th source, as

$$\mathbf{x}_i = [\mathbf{x}_{i,1}^T, \dots, \mathbf{x}_{i,m}^T]^T$$

in which

$$\mathbf{x}_{i,p} = \mathbf{a}(\theta_p) s_{i,p} + \mathbf{n}_{i,p} \quad (73)$$

meaning that

$$\mathbf{y}_i = \sum_{p=1}^m \mathbf{x}_{i,p} = \mathbf{H} \mathbf{x}_i \text{ with } \mathbf{n}_i = \sum_{p=1}^m \mathbf{n}_{i,p} \text{ and } \mathbf{H} = \underbrace{[\mathbf{I}, \dots, \mathbf{I}]}_{m \text{ terms}}$$

Furthermore,

$$\mathbf{x}_{i,p} \sim \mathcal{CN}(\mathbf{a}(\theta_p) s_{i,p}, \boldsymbol{\Sigma}_p)$$

From above one obtains,

$$\log p(X_i|\boldsymbol{\theta}) \propto \mathbf{s}_i^H \mathbf{A}(\boldsymbol{\theta})^H \boldsymbol{\Gamma}^{-1} \mathbf{x}_i - \mathbf{x}_i^H \boldsymbol{\Gamma}^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{s}_i + \mathbf{s}_i^H \mathbf{A}(\boldsymbol{\theta})^H \boldsymbol{\Gamma}^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{s}_i + C$$

in which $\mathbf{\Gamma} = \text{bdiag}(\mathbf{\Sigma}_1, \dots, \mathbf{\Sigma}_m)$, for sake of simplicity, we assume that $\mathbf{\Sigma}_1 = \dots = \mathbf{\Sigma}_m = \frac{1}{m} \mathbf{\Sigma}$. Consequently,

$$\begin{aligned}
Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) &= E_{X_i|\mathbf{y}_i, \boldsymbol{\theta}^{(m)}} \{\log p(X_i|\boldsymbol{\theta})\} \\
&\propto \mathbf{s}_i^H \mathbf{A}(\theta_p)^H \mathbf{\Gamma}^{-1} \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^H \mathbf{\Gamma}^{-1} \mathbf{A}(\theta_p) \mathbf{s}_i + \mathbf{s}_i^H \mathbf{A}(\theta_p)^H \mathbf{\Gamma}^{-1} \mathbf{A}(\theta_p) \mathbf{s}_i + C \\
&\propto (\mathbf{s}_i^H \mathbf{A}(\theta_p) - \tilde{\mathbf{x}}_i)^H \mathbf{\Gamma}^{-1} (\mathbf{s}_i^H \mathbf{A}(\theta_p) - \tilde{\mathbf{x}}_i) + C \\
&\propto \sum_{p=1}^m (s_{i,p} \mathbf{a}(\theta_p) - \tilde{\mathbf{x}}_{i,p})^H \mathbf{\Gamma}_p^{-1} (s_{i,p} \mathbf{a}(\theta_p) - \tilde{\mathbf{x}}_{i,p}) + C
\end{aligned} \tag{74}$$

in which $\tilde{\mathbf{x}}_{i,p} = E_{X_i|\mathbf{y}_i, \boldsymbol{\theta}^{(m)}} \{x_{i,p}\}$. It can be proved that since, $\mathbf{y}_i$ and $\mathbf{x}_i$ are jointly Gaussian (this is because $\mathbf{y}_i = \mathbf{H}\mathbf{x}_i$ with $\mathbf{y}_i$ and $\mathbf{x}_i$ Gaussian), then using classical results from *Anderson 1958* (home work), we have

$$\tilde{\mathbf{x}}_i \triangleq [\tilde{x}_{i,1}^T, \dots, \tilde{x}_{i,m}^T]^T = \mathbf{A}(\boldsymbol{\theta}^{(m)}) \mathbf{s}_i + \mathbf{\Gamma} \mathbf{H}^H (\mathbf{H} \mathbf{\Gamma} \mathbf{H}^H)^{-1} (\mathbf{y}_i - \mathbf{H} \mathbf{A}(\boldsymbol{\theta}^{(m)})) \tag{75}$$

which reduces (using the expressions of $\mathbf{H}$ and $\mathbf{\Sigma}_p$ given above) to

$$\tilde{\mathbf{x}}_{i,p} = \mathbf{A}(\theta_p^{(m)}) \mathbf{s}_{i,p} + \frac{1}{m} \left(\mathbf{y}_i - \sum_{q=1}^m \mathbf{a}(\theta_q^{(m)}) \mathbf{s}_{i,q} \right) \tag{76}$$

Finally, the EM reduces to

EM for multi-source direction of arrival estimation

- *Initialization:* Chose initial estimate $\boldsymbol{\theta}^{(0)}$
- *E-step:* Calculate $Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ from (74) which reduces to the following updates

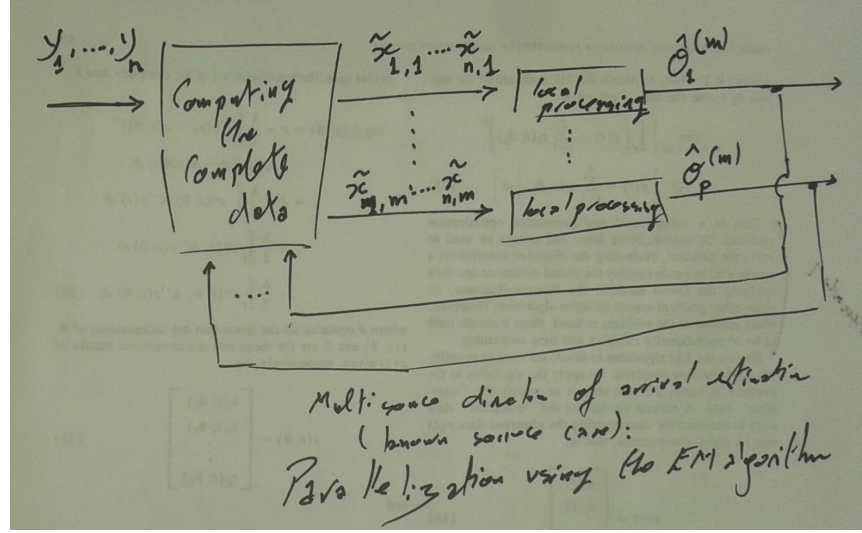
$$\tilde{\mathbf{x}}_{i,p} = \mathbf{a}(\theta_p^{(m)}) \mathbf{s}_{i,p} + \frac{1}{m} \left(\mathbf{y}_i - \sum_{q=1}^m \mathbf{a}(\theta_q^{(m)}) \mathbf{s}_{i,q} \right)$$

- *M-step:*

$$\theta_p^{(m+1)} = \arg \min_{\theta_p} \sum_{i=1}^n (\tilde{\mathbf{x}}_{i,p} - \mathbf{a}(\theta_p) \mathbf{s}_{i,p})^H \mathbf{\Sigma} (\tilde{\mathbf{x}}_{i,p} - \mathbf{a}(\theta_p) \mathbf{s}_{i,p})$$

- *Convergence check:* Compute $l(\boldsymbol{\theta}^{(m+1)})$, if $|l(\boldsymbol{\theta}^{(m+1)}) - l(\boldsymbol{\theta}^{(m)})| > \delta$, for a chosen δ , return to E-step, otherwise end the algorithm

From the above, we can notice that the EM algorithm reduces the original m D optimization problem ($\hat{\boldsymbol{\theta}} = \arg \min \boldsymbol{\theta} l(\boldsymbol{\theta})$) to m subproblems each of 1D optimization problem (74).



B. Other EM variants

1) *The One Step Late: EM for Penalized Likelihood Estimation:* Suppose that we have a prior information about θ (making it a penalized likelihood or maximum a posteriori estimation) and let $J(\theta)$ describing this latter. Consequently, the EM can be formulated as maximizing

$$\theta^{(m+1)} = \arg \max_{\theta'} Q(\theta' | \theta^{(m)}) - \lambda J(\theta') \quad (77)$$

Nevertheless, maximizing (77) can be as difficult as maximizing the original penalized likelihood. This is particularly likely to be so if $J(\theta)$ represents a roughness penalty, when each component derivative of $J(\theta)$ involves several components of θ . To overcome this, *Peter J. Green, 1990* proposed the *one step late EM algorithm* formulated as finding $\theta^{(m+1)}$ which equates the following to zero

$$\left. \frac{\partial Q(\theta | \theta^{(m)})}{\partial \theta} \right|_{\theta=\theta^{(m+1)}} - \lambda \left. \frac{\partial J(\theta)}{\partial \theta} \right|_{\theta=\theta^{(m)}} = 0 \quad (78)$$

Note that the original EM solves the following

$$\left. \frac{\partial Q(\theta | \theta^{(m)})}{\partial \theta} \right|_{\theta=\theta^{(m+1)}} - \lambda \left. \frac{\partial J(\theta)}{\partial \theta} \right|_{\theta=\theta^{(m+1)}} = 0 \quad (79)$$

Solving (78) has the same difficulty as designing the unpenalized EM. For some applications, the OSL works remarkably well, nevertheless, what is lost in comparison with the penalized EM algorithm is the guarantee that the method converges, and in particular that the iteration always increases the penalized

log-likelihood, i.e., $\log f(\mathbf{y}|\boldsymbol{\theta}) - \lambda J(\boldsymbol{\theta})$. In the following, we present the SAGE which overcome this issue.

2) *SAGE : Space-Alternating Generalized Expectation-Maximization Algorithm*: **Definition and properties**

In the following, we recap the SAGE algorithm originally proposed by *Fessler and Hero 1994* in the case of penalized maximum likelihood. First, let us consider the following problem

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \Phi(\boldsymbol{\theta})$$

where

$$\Phi(\boldsymbol{\theta}) \triangleq \log f(\mathbf{y}|\boldsymbol{\theta}) - P(\boldsymbol{\theta}) \quad (80)$$

in which $P(\boldsymbol{\theta})$ represents the penalization term.

Second, let us define the Hidden-data space in the SAGE algorithm procedure defined as:

A random vector X^S is an admissible hidden-data space⁶ w.r.t. to $\boldsymbol{\theta}_S$ for $f(\mathbf{y}|\boldsymbol{\theta})$ if the joint density of X^S and Y satisfies

$$f(\mathbf{y}, \mathbf{x}^S|\boldsymbol{\theta}) = f(\mathbf{y}|\mathbf{x}^S, \boldsymbol{\theta}_{\bar{S}})f(\mathbf{x}^S, \boldsymbol{\theta})$$

This simply means that the conditional distribution $f(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_{\bar{S}})$ must be independent of $\boldsymbol{\theta}_S$. Consequently, X^S must be complete-data space for $\boldsymbol{\theta}_S$ given that $\boldsymbol{\theta}_{\bar{S}}$ is known.

Finally the SAGE algorithm reads:

- Let $\boldsymbol{\theta}^{(0)}$ be an initial parameter estimate.

For $i=0,1, \dots$ (

- Choose an index set $S = S^i$
- Choose an admissible hidden-data space X^{S^i} for $\boldsymbol{\theta}_{S^i}$
- *E-step* Compute

$$\phi^{S^i}(\boldsymbol{\theta}_{S^i}|\boldsymbol{\theta}^{(i)}) \triangleq Q^{S^i}(\boldsymbol{\theta}_{S^i}|\boldsymbol{\theta}^{(i)}) - P(\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}_{\bar{S}^i}^{(i)})$$

in which

$$Q^{S^i}(\boldsymbol{\theta}_{S^i}|\boldsymbol{\theta}^{(i)}) = E_{X^S|Y=\mathbf{y}, \boldsymbol{\theta}^{(i)}} \left\{ \log f(X^S|\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}^{(i)}) \right\}$$

- *M-step*

$$\boldsymbol{\theta}_{S^i}^{(i+1)} = \arg \max_{\boldsymbol{\theta}_{S^i}} \phi^{S^i}(\boldsymbol{\theta}_{S^i}|\boldsymbol{\theta}^{(i)})$$

⁶The set S is an index set (i.e., indexing which parameters of the vector $\boldsymbol{\theta}$ will be considered), and $\bar{S}$ is the complement of S .

- Update $\theta_{\bar{S}^i}^{(i+1)} = \theta_{\bar{S}^i}^{(i)}$ and (optional repeat the *E-step* and *M-step*)

Remark:

- Rather than requiring a strict maximization of $\phi^{S^i}(\theta_{S^i}|\theta^{(i)})$, one could settle simply for local maxima, or for mere increases in $\phi^{S^i}(\theta_{S^i}|\theta^{(i)})$, in analogy with GEM algorithms. These generalizations provide the opportunity to further refine the tradeoff between convergence rate and computation per iteration.
- For a less informative complete data SAGE algorithm has a faster convergence rate, but the maximization step could be harder to solve. We can use the more informative complete data to update the hardest M-step for analytical tractability and the less informative complete data to update the parameter related to the simplest M-step parameters in for faster convergence.

Example: Linear Gaussian Measurement

Let us consider the following model of estimating superimposed linear signals in Gaussian noise

$$\mathbf{y} = \mathbf{a}_1\theta_1 + \cdots + \mathbf{a}_p\theta_p + \mathbf{n} = \mathbf{A}\boldsymbol{\theta} + \mathbf{n} \quad (81)$$

in which the full rank matrix $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_p]$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \text{diag}\{\sigma\}$. Let us consider for simplicity a quadratic penalty, subject to a symmetric nonnegative definite $\mathbf{P}$, reads

$$P(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T \mathbf{P} \boldsymbol{\theta}$$

It is easy to see that the ML gives

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = (\mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{A} + \mathbf{P})^{-1} \mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{y}$$

Nevertheless, for large size of $\mathbf{A}$, the above might be infeasible or at least costly. Let us see how such issue can be handled using the EM and the SAGE algorithms.

First, it is worth mentioning that since the mean of Y is linear in $\boldsymbol{\theta}$, the conventional complete-data for the EM algorithm for this problem is also linear in $\boldsymbol{\theta}$. Here, we restrict our attention to hidden-data spaces X^S whose means are also linear in $\boldsymbol{\theta}$, and for which the conditional mean of Y given X^S is linear in X^S and $\boldsymbol{\theta}_{\bar{S}}$. Considering a general index set S , the natural hidden-data space for $\boldsymbol{\theta}_S$ is

$$\begin{aligned} X^S &\sim \mathcal{N}(\mathbf{B}\boldsymbol{\theta}_S + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}, \mathbf{C}) \\ Y|X^S = \mathbf{x} &\sim \mathcal{N}(\mathbf{G}\mathbf{x} + \bar{\mathbf{G}}\boldsymbol{\theta}_{\bar{S}}, \mathbf{W}) \end{aligned}$$

which is admissible provided the two normal distributions are independent. To be consistent with (81), it is easy to see that $\mathbf{A}_S = \mathbf{G}\mathbf{B}$, $\mathbf{A}_{\bar{S}} = \mathbf{G}\bar{\mathbf{B}} + \bar{\mathbf{G}}$ and $\mathbf{\Sigma} = \mathbf{W} + \mathbf{G}\mathbf{C}\mathbf{G}^T$. Thus,

$$\log f(X^S | \boldsymbol{\theta}_S, \boldsymbol{\theta}_{\bar{S}}^{(i)}) = c + (\mathbf{B}\boldsymbol{\theta}_S + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)})^T \mathbf{C}^{-1} \left(X^S - \frac{1}{2}(\mathbf{B}\boldsymbol{\theta}_S + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)}) \right)$$

Consequently, since X^S and Y are jointly Gaussian and using Bayes rule one obtains

$$\tilde{X}^S = E \left\{ X^S | \mathbf{y}, \boldsymbol{\theta}^{(i)} \right\} = \mathbf{B}\boldsymbol{\theta}_S^{(i)} + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)} + \mathbf{C}\mathbf{G}^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{A}\boldsymbol{\theta}^{(i)})$$

Finally, the E-step reduces to

$$\phi^S(\boldsymbol{\theta}_S, \boldsymbol{\theta}^{(i)}) = (\mathbf{B}\boldsymbol{\theta}_S + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)})^T \mathbf{C}^{-1} \left(\tilde{X}^S - \frac{1}{2}(\mathbf{B}\boldsymbol{\theta}_S + \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)}) \right) - \frac{1}{2} \begin{bmatrix} \boldsymbol{\theta}_S \\ \boldsymbol{\theta}_{\bar{S}}^{(i)} \end{bmatrix}^T \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 \\ \mathbf{P}_2^T & \mathbf{P}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_S \\ \boldsymbol{\theta}_{\bar{S}}^{(i)} \end{bmatrix}$$

and the M-step reduces to

$$\begin{aligned} \boldsymbol{\theta}_S^{(i+1)} &= (\mathbf{F}_{X^S} + \mathbf{P}_1)^{-1} (\mathbf{B}^T \mathbf{C}^{-1} (\tilde{X}^S - \bar{\mathbf{B}}\boldsymbol{\theta}_{\bar{S}}^{(i)}) - \mathbf{P}_2 \boldsymbol{\theta}_{\bar{S}}^{(i)}) \\ &= \boldsymbol{\theta}_S^{(i)} + (\mathbf{F}_{X^S} + \mathbf{P}_1)^{-1} \mathbf{A}_S^T \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{A}\boldsymbol{\theta}^{(i)}) - (\mathbf{F}_{X^S} + \mathbf{P}_1)^{-1} [\mathbf{P}_1 \mathbf{P}_2] \boldsymbol{\theta}^{(i)} \end{aligned}$$

in which $\mathbf{F}_{X^S} = \mathbf{B}^T \mathbf{C}^{-1} \mathbf{B}$.

From above it is easy to deduce the EM and the SAGE algorithms for (81).

EM: For the EM, the complete data space is naturally $S = \{1, 2, \dots, p\}$, $\mathbf{B} = \text{diag}\{\mathbf{a}_k\}$, $\mathbf{C} = \frac{1}{p} \mathbf{I} \otimes \mathbf{\Sigma}$, $\mathbf{G} = (\mathbf{1}^T \otimes \mathbf{I})$ and $\bar{\mathbf{B}} = \bar{\mathbf{G}} = \mathbf{W} = \mathbf{0}$. Consequently, $\mathbf{F}_{X^S} = p \text{diag}\{\mathbf{a}_k \mathbf{\Sigma}^{-1} \mathbf{a}_k\}$ is diagonal. Then, the M-step will be feasible if $\mathbf{P}$ is diagonal, otherwise the use of the EM is useless since we did not overcome the problem of large matrix inversion induced by the ML.

SAGE: Because of the additive form of the data model in (81) it is natural to update each parameter $\boldsymbol{\theta}_k$ individually, i.e., $S^i = \{k\}$. And surprisingly (or not) this will remove the issue of the inversion due to the ML (for all $\mathbf{P}$) and the EM (for non diagonal $\mathbf{P}$). Also, it is of interest to associate all of the noise covariance with the signal vector $\mathbf{a}_k$ (see the discussion in Appendix C of *Fessler and Hero 1994*). Consequently, one has

$$\begin{aligned} X^{S^k} &\sim \mathcal{N}(\mathbf{a}_k \boldsymbol{\theta}_k, \mathbf{\Sigma}) \\ Y &= X^{S^k} + \sum_{j \neq k} a_j \boldsymbol{\theta}_j \end{aligned}$$

in this case $\mathbf{B} = \mathbf{a}_k$, $\mathbf{C} = \mathbf{\Sigma}$, $\bar{\mathbf{B}} = \mathbf{W} = \mathbf{0}$, $\mathbf{G} = \mathbf{I}$, $\bar{\mathbf{G}} = [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_p]$ and $\mathbf{F}_{X^{S^k}} = \mathbf{a}_k$.

This leads naturally to the following algorithm

- Initialize: $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{A}\boldsymbol{\theta}^{(0)}$

- for $i = 0, 1, \dots$,

$$k = 1 + (i \bmod p), S = \{k\},$$

$$\boldsymbol{\theta}_k^{(i+1)} = \boldsymbol{\theta}_k^{(i)} - ((\mathbf{a}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{a}_k + [\mathbf{P}]_{k,k})^{-1}) \left([\mathbf{P}]_{k,:} \boldsymbol{\theta}^{(i)} - \mathbf{a}_k^T \boldsymbol{\Sigma} \hat{\mathbf{e}} \right)$$

$$\hat{\mathbf{e}} = \hat{\mathbf{e}} + (\boldsymbol{\theta}_k^{(i+1)} - \boldsymbol{\theta}_k^{(i)}) \mathbf{a}_k$$

$$\boldsymbol{\theta}_k^{(i+1)} = \boldsymbol{\theta}_j^{(i)}, \quad j = 1, \dots, k-1, k+1, \dots, p$$

Note that unlike the EM algorithm, the SAGE algorithm circumvents the need to invert $\mathbf{P}$ by performing a sequential update, so a non-diagonal smoothness penalty $\mathbf{P}$ is entirely feasible.

REFERENCES

- [1] A. Hippert-Ferrer, M. N. El Korso, A. Breloy, and G. Ginolhac, “Robust low-rank covariance matrix estimation with a general pattern of missing values,” *Signal Processing*, vol. 195, p. 108460, June 2022.
- [2] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 3rd ed. Wiley, 2019.
- [3] J.-P. Delmas, M. N. El Korso, F. Pascal, and S. Fortunati, *Elliptically Symmetric Distributions in Signal Processing and Machine Learning*. Springer Nature, Dec. 2024.
- [4] A. Hippert-Ferrer, A. Sportisse, A. Javaheri, M. N. El Korso, and D. P. Palomar, “Missing data in signal processing and machine learning: Models, methods, and modern approaches,” *IEEE Signal Processing Magazine*, vol. 43, no. 4, 2026.
- [5] B. Delyon, M. Lavielle, and E. Moulines, “Convergence of a stochastic approximation version of the EM algorithm,” *The Annals of Statistics*, vol. 27, no. 1, pp. 94–128, 1999.
- [6] R. J. A. Little, “Robust estimation of the mean and covariance matrix from data with missing values,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 37, no. 1, pp. 23–38, 1988.
- [7] D. E. Tyler, “A distribution-free M-estimator of multivariate scatter,” *The Annals of Statistics*, vol. 15, no. 1, pp. 234–251, 1987.
- [8] M. Haardt, M. Pesavento, F. Roemer, and M. N. El Korso, “Subspace methods and exploitation of special array structures,” in *Academic Press Library in Signal Processing*. Academic Press, 2017, vol. 3, pp. 651–717.
- [9] B. Meriaux, C. Ren, A. Breloy, M. N. El Korso, and P. Forster, “Matched and mismatched estimation of kronecker product of linearly structured scatter matrices under elliptical distributions,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 603–616, 2021.